UNIVERSIDADE DE LISBOA

FACULDADE DE CIÊNCIAS

DEPARTAMENTO DE ESTATÍSTICA E INVESTIGAÇÃO OPERACIONAL

INTRAGENIC INITIATION IN SETD2 DEFICIENT CELLS

Miguel Maria das Neves Sousa Pereira

Trabalho de Projecto

MESTRADO EM BIOESTATÍSTICA

2014

UNIVERSIDADE DE LISBOA

FACULDADE DE CIÊNCIAS

DEPARTAMENTO DE ESTATÍSTICA E INVESTIGAÇÃO OPERACIONAL

INTRAGENIC INITIATION IN SETD2 DEFICIENT CELLS

Miguel Maria das Neves Sousa Pereira

Trabalho de Projecto

MESTRADO EM BIOESTATÍSTICA

Trabalho de projecto orientado pela Professora Doutora Lisete Sousa e pela Professora Doutora Ana Rita Grosso.

2014

Acknowledgements

To my mentors, Lisete Sousa and Ana Rita Grosso, I owe my deepest appreciation. You were always helpful, supportive and, without your guidance, I would not have surpassed all the hurdles I came across doing this work. You were always very understanding, specially with my working schedule, and I cannot thank you enough for that.

To all my running buddies for being such great people with who I share a special and inexplicable connection. Thank you for all the moments spent out in the trails that are a source of strength to keep working and, most of all, thank you for showing me that the word *limit* makes no sense in any aspect of our lives.

Last but not least, I wish to thank my mother for always understanding and supporting my passion for science and without whom I would not have gotten to this point my career.

Miguel, October 2014

Abstract

The advent of high-throughput Next Generation Sequencing technologies that generate information about the genome, transcriptome and epigenome has created a demand for the development of statistical approaches to detect biological phenomena that occur on a molecular level. One of those phenomena is intragenic initiation, which corresponds to transcription initiation in an exon downstream of the first.

The aim of this thesis is to apply statistical methods to identify intragenic initiation and to use these to study the effect of down-regulation due to mutation in the SETD2 gene, a putative tumor suppressor gene in clear cell renal cell carcinoma.

We analyzed count data from a RNA-seq experiment, a Next Generation Sequencing method to obtain the transcriptome of a cell, to detect intragenic initiation in six cell lines: two controls and four cell lines with loss of function mutations in the SETD2 gene. Our approach was based on transforming the data into proportions and comparing pairs of proportions either using the two proportions comparison test along with the Benjamini-Hochberg procedure to correct for multiple testing or the Marascuilo procedure, a method that performs every pair-wise comparison in an experimental unit and incorporates correction for multiplicity.

Our results showed that the two proportions comparison method was not able to effectively detect intragenic initiation since very few genes were detected that had no relation with genes detected by the Marascuilo procedure and other published data. The Marascuilo procedure, on the other hand, detected 1304 genes with approximately 300 genes per mutant sample. There was 50% overlap between at least two mutant cell lines, which suggests that the method is consistent.

We conclude that the Marascuilo procedure seems to be a method that can be applied to the detection of intragenic initiation and allows detection

of this phenomenon in each of the cell lines individually.

Keywords: Next Generation Sequencing (NGS), Intragenic initiation, Proportion comparison, Multiple testing, Marascuilo Procedure.

Resumo

Nos últimos anos observou-se um enorme desenvolvimento no campo da sequênciação genética com o desenvolvimento das plataformas de elevada produção de *Next Generation Sequencing* (NGS). Com a tecnologia de NGS é possível sequenciar um genoma ou um transcriptoma por completo em apenas horas ou dias, o que constitui um avanço importante quando comparado com os métodos de sequenciação de Sanger. A plataforma de NGS é baseada na fragmentação e amplificação através de PCR de DNA ou RNA em pequenos segmentos, denominados *reads*, e na seleção das *reads* que alinham com um genoma de referência. Estas, denominadas *reads* mapeadas , são selecionadas para análise e estudo de fenómenos a nível molecular celular.

O desenvolvimento destas tecnologias foi acompanhado da necessidade de desenvolver ferramentas de bioinformática para analisar dados de NGS. Estas ferramentas são indispensáveis para traduzir e estudar fenómenos genéticos a partir dos dados não processados obtidos a partir dos aparelhos de sequenciação. Um aspecto importante da NGS é a possibilidade de estudo de fenómenos a nível do genoma e do transcriptoma *versus* ao nível de genes e proteínas individualmente. Neste trabalho em particular, é de salientar a possibilidade de estudar o fenómeno de iniciação intragénica da transcrição, que corresponde à iniciação da transcrição de DNA em RNA mensageiro num exão que não o primeiro (que corresponde ao local usual de início da transcrição).

Recentemente, o gene SETD2 foi identificado como sendo um possível gene supressor de tumor em linhas celulares de carcinoma renal de células claras. Este gene codifica uma histona metiltransferase responsável pela trimetilação da lisina 36 da histona H3 (H3K36me3). É já sabido que a ausência de expressão de SETD2 resulta em instabilidade de microssatélites e num aumento da taxa de mutação, motivo pelo qual se associa a reduzida expressão de SETD2 ao cancro. Adicionalmente, a H3K36me3 mediada pelo SETD2 parece estar associada a alteração dos padrões de *splicing* e a um

aumento da iniciação intragénica.

O objectivo deste trabalho é aplicar métodos estatísticos para identificar iniciação intragénica da transcrição e usar os mesmos para estudar o efeito da mutações de SETD2 neste fenómeno usando linhas celulares de carcinoma renal de células claras.

Neste trabalho foram aplicados dois métodos de deteção de iniciação intragénica que foram aplicados em dados provenientes de seis linhas celulares de carcinoma renal de células claras: quatro linhas celulares com mutações *loss of function* do gene SETD2 (AB, ER, MF e FG2) e duas linhas-controlo (Caki1 e Caki2). Primeiro, foi efetuado o processamento dos dados originados pelo aparelho de sequenciação para obter dados de contagens relativos ao números de *reads* que alinharam com cada exão. Estes dados foram transformados em proporções tendo como base as contagens de *reads* e numa estimativa do número de *reads* que existiriam se todos os exões tivessem o mesmo nível de expressão e se comportassem como um gene activo. O objectivo foi comparar a expressão entre todos os pares de exões contiguous em cada gene e seleccionar os genes com um nível de expressão superior por parte de um exão que não o primeiro, o que sugere a existência de iniciação intragénica da transcrição.

Para cada linha celular, obteve-se um vector com as proporções de expressão de cada exão organizadas por gene e foram aplicados dois métodos estatísticos que podem ser usados para comparação múltiplas proporções: (1) o teste de comparação de duas proporções e (2) o método de Marascuilo. No método (1) foram efetuadas todas as comparações dois-a-dois entre os pares de exões contíguos de cada gene e, um vez que se tem comparações múltiplas, os valor-p obtidos foram ajustados usando o procedimento de Benjamini-Hochberg que controlo a proporção de verdadeiras hipóteses nulas em cada gene. No método (2) foi aplicado um método que efetua todas as comparações dois-a-dois entre os exões de cada gene e selecionados os pares de interesse (ou seja, os pares de exões contíguos). Este método testa mais pares do que o necessário levando a um maior número de comparações. Isto cria um viés na direção da hipótese nula o que faz deste procedimento um método conservador. No entanto, o método de Marascuilo tem a vantagem de incorporar a correção para testes múltiplos não sendo necessário aplicar um outro método para efetuar esse ajuste.

Na nossa abordagem, um gene tem iniciação intragénica da transcrição se satisfizer os seguintes critérios: (1) apresentar um exão *downstream* com maior nível de expressão que o primeiro exão (ou primeiros exões), (2) o

primeiro exão diferencialmente expresso positivamente encontrar-se nos primeiros 40% dos exões do gene e (3) mantiver os nível de expressão em 50% dos exões *downstream* do primeiro exão diferencialmente expresso.

Utilizando o nosso algoritmo de processamento de dados foram identificados 42233 genes, excluindo isoformas, dos quais 13667 (≈31.7%) foram excluídos por serem compostos por apenas quatro exões ou menos. Optou-se por excluir estes genes *a priori* visto ser impossível os mesmos obedecerem aos critérios de iniciação intragénica estabelecidos.

Os nossos resultados mostraram que o teste de comparação de duas proporções juntamente com o procedimento de Benjamini-Hocherg não consegiu identificar um número satisfatório de genes. Adicionalmente, os genes identificados não apresentavam qualquer concordância com os detectados com o método de Marascuilo e com dados previamente publicados. Foi colocada a hipótese deste aspecto se dever à sensibilidade para este método detectar pequenas flutuações de expressão por ser um método menos conservador do que o método de Marascuilo, bem como ao facto dos critérios para considerer iniciação intragénica serem estritos.

Por outro lado, o método de Marascuilo, detectou 1304 genes com iniciação intragénica tendo identificado cerca de 500 genes em cada amostra. Destes, ≈300 genes eram específicos das linhas celulares mutadas quando elimiandos os genes concordantes entre cada uma destas linhas e o controlo Caki1. Pelo contrario, quando eliminados os genes concordantes entre as amostras Caki1 e Caki2, foram detectados apenas 208 genes nesta última amostra, o que vem ao encontro da previsão de que a mutação no gene SETD2 aumenta a inicição intragénica.

Com este trabalho pode-se concluir que o método de Marascuilo pode ser usado como uma ferramenta para detetar iniciação intragénica. Este método deteta um menor número de genes que um método menos convervador baseado no teste exato de Fisher previamente descrito, mas apresenta a vantagem de identificar numa amostra sem necessitar de comparar com o controlo, ou seja, é capaz de detectar a iniciação intragénica basal numa linha cellular controlo, por exemplo. Adicionalmente, este método parece ser preciso na quantificação de iniciação intragénica tendo-se detectado uma concordância de 50% entre pelo menos duas linhas mutadas, o que contrasta com uma concordância de 22% no método previamente descrito.

Palavras-chave: Next Generation Sequencing (NGS), Iniciação intragénica, Comparaçõo de proporções, Testes múltiplos, Método de Marascuilo.

Contents

List of Figures

List of Tables

Preface

In recent years important developments have occurred in the field of genetic sequencing with the development of high-throughput Next Generation Sequencing (NGS) technologies. With NGS technology it is possible to sequence a whole genome or transcriptome in hours or days, which constitutes major breakthrough when compared with Sanger sequencing-based methods. The NGS framework is based on fragmenting and amplifying millions of DNA or RNA pieces, called reads, and selecting the reads that align with a reference genome. The reads that align (called mapped reads) are selected for analysis and study of molecular biology phenomena.

The development of these technologies was accompanied the need to develop bioinformatics tools to analyze NGS data. These tools are indispensable to perform research since they work as translators of the raw data produced by the sequencing machines. An important aspect of Next-Generation sequencing is that it permits the study of biological phenomena that occur at the genome or transcriptome level. In particular to this work, it permits the study of the phenomenon of intragenic transcription initiation, which is the initiation of transcription of DNA into mRNA starting at an exon other than the first (the usual location for transcription to start).

Recently, the SETD2 gene has been identified as putative tumor suppressor in cell lines of clear cell renal cell carcinoma, the most common type of renal cancer. This gene codes for a histone methyltransferase that is responsible for the trimethylation of lysine 36 of histone H3 (H3K36me3). It is known that lack of expression of SETD2 in cells results in microsatellite instability and an elevated mutation rate, which explains the association between loss of function of SETD2 and cancer. Moreover, SETD2-mediated H3K36me3 seem to be associated with changes in splicing patterns and in an increase in intragenic initiation.

The aim of this thesis is to apply statistical methods to identify intragenic

initiation and to use these to study the effect of down-regulation due to mutation in the SETD2 gene in this phenomenon.

In this manuscript we applied two methods to detect intragenic initiation and applied them to six clear cell renal cell carcinoma cell lines: four cell lines carrying loss-of-function mutations in the SETD2 gene (AB, ER, MF and FG2) and 2 normal controls (Caki1 and Caki2). First, we processed the raw data from the RNA-seq experiment in order to obtain read counts for each exon of each cell line. Then, we transformed the count data into proportions based on the reads counts of each exon and on an estimate of the reads counts considering that every exon of every gene has equal expression and behaves as an active gene. Our goal was to compare the expression between every pair of contiguous exons and select the genes that showed significantly higher expression of a downstream exon that suggested that transcription initiation had occurred at an exon other that the first.

For each cell line, we obtained a vector with exon expression proportions organized by gene and applied two statistical methods that can be used to compare multiple proportions: (1) The two proportions comparison test and (2) the Marascuilo procedure. In (1), we performed all comparisons between contiguous exons and, since we incur in a problem of multiple testing, adjusted the obtained p-values using the Benjamini-Hochberg procedure, a method that controls the False Discovery Rate or the proportion of true null-hypothesis. In (2), we applied a procedure that performs every pair-wise comparison between exons in a gene and selected the pairs of consecutive exons. Testing more pairs than the pairs of interest biases the results towards the null-hypothesis thus making this a conservative method of detection. The Marascuilo procedure has the advantage of incorporating the correction for multiple testing in itself thus not requiring the concomitant use of a method of correction for multiple testing.

In our approach a gene has intragenic initiation if it satisfies one of the following criteria: (1) a downstream exon is more expressed than the first exon/s, (2) the first expressed exon is within the first 40% of the exons in the gene and (3) the expression of the exons downstream of the first in stable in at least 50% of those exons. Using our data processing algorithm, we identified 42233 genes, excluding isoforms, of which 13667 genes ($\approx$31.7%) were excluded because they were composed of less than 5 exons. We opted to exclude these because with our criteria genes with less that 5 exons would never show intragenic initiation. Our results showed that the two proportions comparison method was not able to effectively detect intragenic initiation since very few genes were identified. Moreover, these had no concordance

with the genes detected by the Marascuilo procedure and other published data. We hypothesize that this was due to the number of fluctuations in expression that this method, being less conservative than the Marascuilo procedure, can detect allied to the relatively strict criteria to consider that a gene showed intragenic initiation. The Marascuilo procedure, on the other hand, identified 1304 genes with approximately 500 genes per sample. Of these, ≈300 genes were specific to the mutant cell lines when we eliminated the genes that we belonged to the intersection between the control Caki1 and each of the mutant cell lines. In contrast, when we eliminated the genes detected in both Caki1 and Caki2, we identified only 208 genes in Caki2, which suggests that mutations in SETD2 are associated with an increase in intragenic initiation.

With this work we can conclude that the Marascuilo procedure can be used as a tool to detect intragenic initiation. It detects less genes than a previously described less conservative method using Fisher′s exact test but has the advantage over the latter of detecting genes in a sample without requiring it to be compared with a control. In this sense, it detects baseline intragenic initiation. Additionally, it seems to be an accurate method in quantifying this phenomenon since it detected a 50% concordance between at least two mutant samples.

Chapter 1

Biological Background

This chapter is meant to introduce the biological framework behind the problem we want to approach. It provides an overview of concepts of cell and molecular biology inspired by Cooper and Hausman′s textbook ″The Cell, a Molecular Approach″ (2007) and the state of the art regarding renal cell carcinoma and the SETD2 gene, whose effects in the former cells we want to elucidate. It will also provide a basis of understanding of Next Generation Sequencing, the technology used to generate the data that will allow us to answer our proposed question. To finalize, we will present the aims of this thesis.

1.1 The genome and genes

The genome constitutes the entirety of the hereditary information in a cell. According to each organism, it is constituted either by deoxyribonucleic acid (DNA) or ribonucleic acid (RNA). In eukaryotes, the genome is composed of DNA located in the nucleus of the cells and it is normally divided in several pieces called chromosomes. Each chromosome is further divided into genes that constitute the basis of the hereditary information of a cell. Each gene serves as a template to synthesize RNA through a highly regulated process known as transcription that consists of a series of steps to obtain processed RNA (also known as messenger RNA or mRNA). The mRNA then migrates to the cytoplasm where will be "read" by the endoplasmic reticulum to produce proteins, a process known as translation. Even though, the products of gene expression are the sole responsibles for the phenotype of a cell the

remainder of the genome (also known as non-coding DNA as opposed to the coding DNA corresponding to genes) plays a very important role in the regulation of gene expression. In particular, the non-coding DNA is constituted by regulatory sequences that modulate gene expression (e.g. transcription activators and inhibitors). Moreover, each chromosome is not simply constituted by DNA (or RNA). Despite the differences among different organisms, eukaryotic chromossomes are also constituted by proteins, called histones, with the DNA wrapped around them forming a coil-shaped structure. Histones play a very important role in the regulation of gene expression by altering their structure in response to certain stimuli (e.g. the methylation of a particular aminoacid) thus promoting or repressing transcription. In the same way the genome constitutes the whole of a cell's genetic information, the transcriptome is the group of all mRNAs in each cell according to its pattern of gene expression and the proteome the group of proteins produced by a cell from mRNA. The latter is normally considered a direct product of the expression of the transcriptome.

1.2 Transcription

Transcription starts with the synthesis of RNA from a DNA template. This is mediated by an enzyme called RNA polymerase that binds to DNA and uses it as a template to synthesize mRNA. RNA polymerase recognizes a particular region of the gene called the promoter where it binds to start transcription. Promoters are regions normally located upstream of the gene that contain specific sequences recognized by RNA polymerase.

After transcription, the mRNA is processed to its final form before migrating to the cytoplasm. In particular, parts of the mRNA, the introns, are removed to obtain the final mRNA that will be translated (Figure 1.1). The mRNA fragments that remain, the exons, are ligated to form processed mRNA. The process of removing introns in named splicing and is a form of generating transcriptional diversity. Several genes show more the one pattern of intron removal. Therefore, one gene can originate more than one mRNA and consequently, different proteins. In general, splicing occurs with the removal of one particular set of introns. However, under certain circumstances, alternative splicing (removal of a different set of introns) can occur thus forming alternative spliced isoforms.

Another pathway of generation of transcript diversity is intragenic initiation (ITI). In this situation, RNA polymerase binds to a cryptic promoter

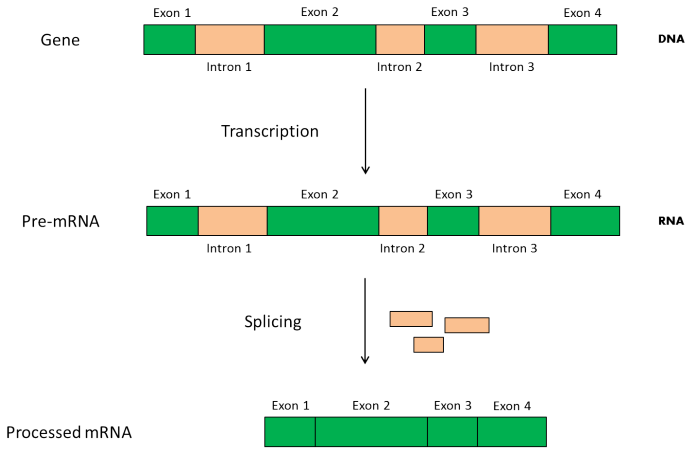

Figure 1.1: Transcription and mRNA processing. RNA polymerase synthesizes pre-mRNA from DNA. Pre-mRNA is further processed and introns are removed to form processed mRNA that will migrate to the cytoplasm and will serve as a templante to protein synthesis in the endoplasmic reticulum.

site, that is, a promoter-like site in the gene where RNA polymerase can bind to initiate transcription (Pattenden *et al.*, 2010). Although these promoters allow transcription initiation, it is known that their structure is different from normal promoters and the exact mechanism of action of RNA polymerase in this case is yet to be elucidated. Nevertheless, it has been established in yeast that mutations in certain genes promote intragenic initiation. Specifically, mutation of components of the Set2-Rpd3S pathway results in cryptic transcription initiation within the coding region of approximately 30% of yeast genes (Pattenden *et al.*, 2010).

1.3 DNA sequencing and Next Generation Sequencing

Sequencing is the process of determining the nucleotide order in a sample of nucleic acids (DNA or RNA). DNA sequencing first started in the early 1970s with the first sequences being obtained using two-dimensional chromatography. In 1973, Maxam and Gilbert sequenced the lac operator (24 basepairs) using a method known as wandering spot analysis (Maxam and Gilbert, 1976). In 1977, Sanger sequenced the first whole DNA genome from the bacteriophage ϕX174 (Sanger *et al.* , 1977(1)). Sanger developed a rapid

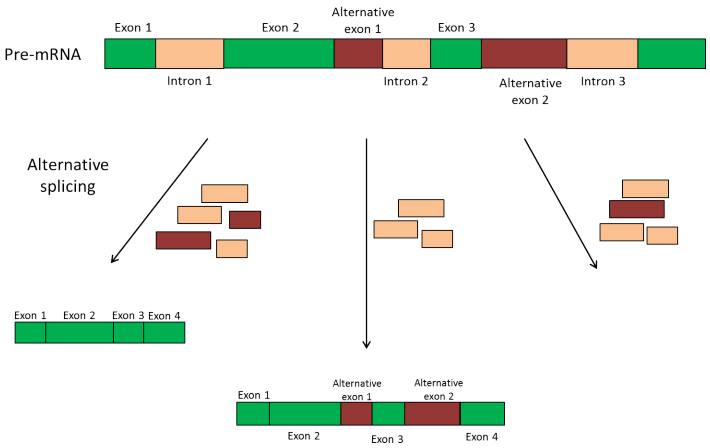

Figure 1.2: Alternative splicing. Some exons are alternatively spliced, that is, they can remain in the mRNA or be removed via splicing. This mechanism allows the production of different transcripts and, consequently, proteins, meaning that one gene can code for more than one proteins. In general, alternative splicing is dependent on the presence of slicing factors that mediate the removal of each set of introns and alternative exons.

DNA sequencing method based on chain-terminating inhibitors (Sanger *et al.* , 1977(2)). Later in 1977, Maxam and Gilbert introduced a novel method based on chemical modification of DNA and cleavage at specific sites (Maxam and Gilbert, 1977). Maxam and Gilbert's methods become more popular at first but were supplanted by Sanger's method, which was less technically demanding and employed less radiation and toxic chemicals than the former (Saccone and Pesole, 2005).

Sanger's method was the basis for the methods that followed. These had the advantage of faster (and cheaper) sequencing due to the possibility of parallelization. In the late 1980s the first semi-automated and automated sequencing machines were developed and this jump-started the sequencing of whole genomes. In 1995, the first free-living organism genome from *Haemophilus influenzae* was sequenced (Fleischmann *et al.* , 1995). Later, in 2001, a draft of the human genome was obtained using the shotgun *de novo* sequencing approach of the Sanger method (Figure 1.3a) and, by 2004 the complete human genome had been sequenced.

In the mid to late 1990s, Next Generation Sequencing (NGS) was introduced. The "next-generation" methods allowed whole genome sequencing significantly faster than the Sanger's method-based approaches thus opening

the possibility of sequencing whole chromosomes and whole genomes in days. The next-generation methods are very diverse but their general workflow is similar (Figure 1.3b): DNA is randomly fragmented and common adapter sequences are ligated to the ends of each fragment. The fragments are then spatially separated and will undergo several cycles of PCR amplification. The amplified fragments will cluster around the original fragment thus forming "polonies". The sequence is obtained by alternating cycles of nucleic acid synthesis via a polymerase or a ligase and imaging to obtain data on each added nucleotide.

The data used in this work was obtained using the Illumina® Genome Analyzer HiSeq2000. This method is a sequencing by synthesis technology (SBS) where the adapters are attached by a flexible linker in a plane structure, known as the flow cell. The DNA fragments are amplified by bridge PCR (Figure 1.4) and clusters of each fragment are formed in the cell. The synthesized single-stranded DNA fragments are called reads. This method forms reads between 75 base pairs (bp) and 100bp and allows mapping of the whole genome or sequencing of the transcriptome, for example.

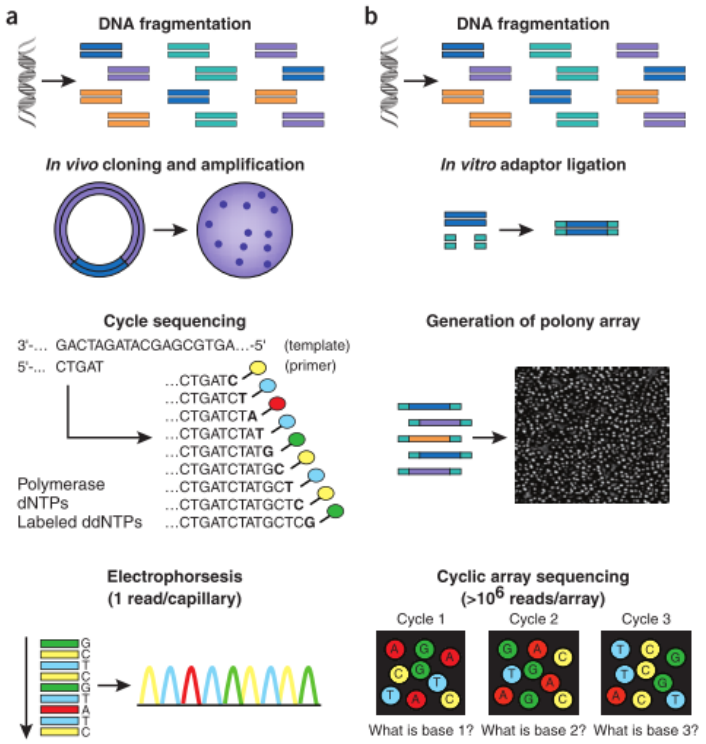

Figure 1.3: Shotgun Sanger sequencing and next-generation sequencing. (a) In Sanger's sequencing genomic DNA is fragmented, cloned to a plasmid vector and used to transform *E. coli*. Each colony is used in a reaction where plasmid DNA is isolated and subject to a "cycle sequencing" reaction where cycles of template denaturation, primer binding and primer elongation take place. The sequence is obtained by high-resolution electroforetic gel separation of the DNA fragments synthesized from the genomic DNA templates. An alternative to shotgun Sanger sequencing that is targeted to a segment of the genome is based on the creation of DNA fragments and the PCR amplification using specific primers for the region of interest. Sanger's sequencing method can then be applied. (b) In next-generation sequencing methods, genomic DNA is fragmented and ligated to a set of adaptors that can bind to an array where each DNA fragment will be spatially separated. These fragments are then subject to several cycles of amplification generating PCR colonies or "polonies". Each fragment undergoes alternating cycles of DNA synthesis with fluorescent-labeled nucleotides and imaging to detect each inserted nucleotide. This approach allows the parallel sequencing of all the fragments in one single array. Although not all next-generation methods are based on a planar array, the other methods are all based on the spatial separation of the fragments and therefore the same principles apply (figure adapted from Shendure and Hanlee, 2008).

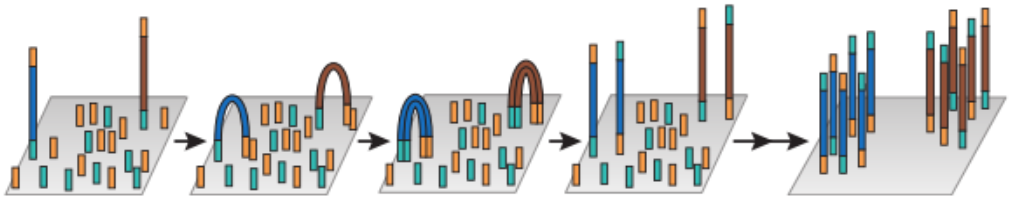

Figure 1.4: Bridge PCR. Single stranded genomic DNA fragments bind one of the adapter sequences, bend (or form a bridge) and ligate the opposite end to a free adaptor in the flow cell. DNA synthesis (amplification) takes place in this setting. After amplification the bridge is dismantled by DNA denaturation and separation of each strand. Several cycles of amplification and denaturation lead to the formation of single-stranded DNA fragment clusters (reads), since the adapters are in fixed locations in the array.

The advent of NGS technologies brought more applications than simple DNA sequencing. These techniques allow determination of the transcriptome (a method known as RNA-sequencing or RNA-seq), small noncoding RNA (ncRNA) discovery and profiling, detection of histone modifications (ChIP-seq) and chromatin interactions (ChIA-PET), DNA methylation profiling (MeDIP-seq) and mapping of nucleosome protected DNA (MNase-seq), among others (Kim and Yu, 2012).

The data provided for our analysis was RNA-seq data since the biological goal is to determine the effect of the gene SETD2 in transcriptome alterations through intragenic transcription initiation.

1.3.1 RNA-seq data analysis

RNA-seq is a NGS method that allows identification and quantification of all the transcripts in a cell (transcriptome) (Oshlack et al. 2010). Nowadays, it is a widely used tool in transcriptomics with several advantages over microarrays (Ozsolak *et al.* , 2011) and Sanger sequencing-based methods like expressed sequence tag (EST) sequencing (Nagaraj et al. 2008). It has advantages over microarrays because it does not depend on prior knowledge of the organism's genome, yields significantly less background noise and has a single-base resolution (Oshlack *et al.* , 2010, Wang *et al.* , 2008 and Marguerat *et al.* , 2008). In addition, RNA-seq is advantageous relative to EST sequencing which is relatively low-throughput, expensive and less precise since only a portion of the transcript is analysed (Wang *et al.* , 2008). RNA-

seq is based on the general framework of NGS or high-throughput technologies. It starts with isolation and purification of RNA from a cell line (Figure 1.5). The RNA is used as a template to synthesize cDNA (complementary DNA, DNA synthesized from processed mRNA therefore not containing introns) with adaptors attached to one or both ends. Each cDNA fragment is amplified and sequenced in a high-throughput manner, as previously described, to obtain short sequences (reads) of one (single-end sequencing) or both ends (pair-end sequencing) of the fragment. This method produces reads with 30-400 bp (75-100 bp with Illumina's HiSeq 2000 technology) (Wang *et al.* , 2008). The number of reads produced with a certain sequence will be proportional to the number of mRNA molecules present in the cell thus allowing inference on a cell's genetic expression profile (Morozova and Marra, 2008).

The analysis of RNA-seq data can be highly variable according to the biological question. RNA-seq is mostly used for gene expression profiling between a wild-type and a mutant samples but it can also be applied to detect differential allelic expression (Wagner *et al.*, 2010 and Wang *et al.*, 2008), alternative splicing (Pan *et al.*, 2008 and Sultan *et al.*, 2008) and fusion genes (Ozsolak *et al.*, 2011 and Maher *et al.*, 2009).

After processing the sample, the high-throughput sequencing platform generates a file consisting of short sequences (reads) and an associated quality score (Oshlack et al. 2010). The analysis steps that follow are illustrated in Figure 1.6. First, the sequences are aligned to a reference genome or transcriptome. Second, the aligned reads are assembled (or summarized) to obtain counts of the number of reads that were mapped to each gene, exon or transcript, according to the goal of the experiment. Third, the data are normalized to enable accurate comparison of expression between and within samples. Finally, statistical analysis of differential expression is performed to attain a list of genes/exons/transcripts with associated p-values from which a biological insight can be taken.

Normalization is an essential step in the analysis of RNA-seq data and there are several methods and packages available that standardize read counts. Normalization methods can be based on Total Counts (TC), on the Upper Quartile (UQ), Median (Med) or on the calculation Reads Per Kilobase per Million mapped reads (RPKM) (Dillies *et al.*, 2013). Additionally, the are packages whose function is to perform normalization like `EDASeq` (Risso *et al.*, 2011) and packages that incorparate normalization tools like `DESseq` and `edgeR` (Anders and Huber, 2010). These packages are open source and available at Bioconductor (`http://www.bioconductor.org`). In our case, we

opted to use the simple method of converting read counts to RPKM, specially because in previous with the data used the same normalization method was applied.

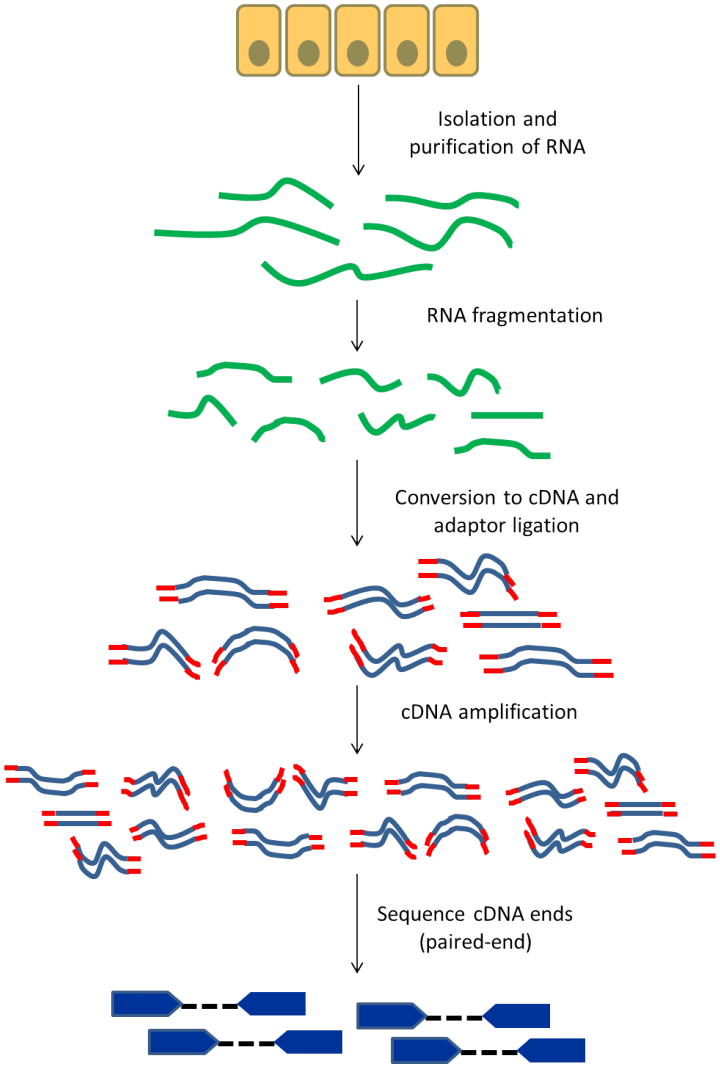

Figure 1.5: RNA-seq experiment. After isolation of RNA from a cell, the mRNA molecules are fragmented and used as templates to synthesize cDNA which is attached to adaptors and amplified (the cDNA is used as the genomic DNA fragments in a standard DNA sequencing experiment). The cDNA extremities are sequenced, generating short reads. In a single-end sequencing experiment only one end of the cDNA is sequenced (since only one adaptor is attached to the cDNA) whereas in a paired-end experiment, both ends are sequenced as represented in the figure.

There are several available software products (based on different approaches which go beyond the scope of this manuscript) to perform each of the described steps. Given the potential and quality of NGS technologies, the analysis of high-throughput data is hot topic in biostatistics and bioinformatics research with new techniques and software's being developed at a very fast rate. In chapter 2, we will further discuss some techniques of statistical analysis of differential expression (DE) used in this setting and present the methods used in our analysis.

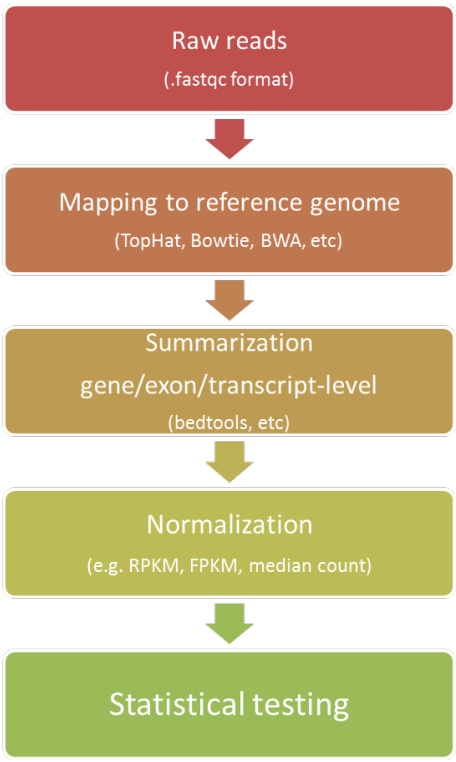

Figure 1.6: RNA-seq workflow. The output of the sequencer is a *.fastqc* file that contains the sequences of each read. The first step is to align the reads to a reference genome or transcriptome available in online databases using software's like Tophat, Bowtie or BWA that use different methods of alignment. Second, the data is summarized at gene, exon or transcript level in order to obtain count data regarding the number of reads that aligned with each unit. Bedtools is on open-source available software that transforms *.sam* or *.bam* files that contain aligned reads in count data files in the *.bed* format. These data can be analyzed using statistical software to test for DE, for example.

1.4 Clear cell renal cell carcinoma and the SETD2 gene

Renal cell carcinoma (RCC) is the most frequent type of kidney cancer. It originates in the epithelial cells of the proximal convoluted tube of the kidney and corresponds to approximately 3% of all human malignancies (Jemal A, Cancer Statistics 2009). According to the World Health Organization there are three histological subtypes of renal cell carcinoma: clear cell RCC (ccRCC), papillary RCC (pRCC) and chromophobe RCC (chRCC) (Eble *et al.* , 2004). The ccRCC is the most frequent subtype, corresponding 80%-90% of all renal cell carcinomas.

Several genetic alterations have been associated with RCC. In particular, mutations in the Von Hippel-Lindau (VHL) gene, chromosome 3p translocation and mutations in the succinate dehydrogenase B (SDHB) gene have been shown to play a role in the pathogenesis of ccRCC (Cancer Genome Atlas Research Network, 2013). Mutations in the VHL tumor suppressor gene account for 60% of the sporadic ccRCC and are also associated with the Von Hippel-Lindau disease, an autosomal dominant genetic disorder characterized by retinal angiomas, hemamgioblastomas of the central nervous system, pheochromocytomas and ccRCC. Moreover, translocations in chromosome 3p have been associated with ccRCC in part because the VHL gene is involved. However, translocations involving other regions of chromosome 3p, namely 3p21 also seem to be involved in the pathophysiology of ccRCC thus suggesting the presence of other tumor suppressor genes in that region. Recently, Duns et al. have identified the SETD2 gene as putative tumor suppressor in cell lines of ccRCC with 3p21 copy number loss. The SETD2 gene, the human counterpart of the Set2 gene from yeast, codes for a histone methyltransferase that is nonredundantly responsible for the trimethylation of lysine 36 of histone H3 (H3K36me3). Lack of expression of SETD2 in cells results in microsatellite instability and an elevated mutation rate which explains the association between loss of function of SETD2 and cancer (Li *et al.* , 2013).

SETD2-mediated H3K36me3 is also associated with changes in transcription. Misteli *et al.* showed that histone modifications like H3K36me3can alter splicing patterns. More recently, it has been shown that SETD2 modulates FACT (FAcilitates Chromatin Transcription) and inhibits initiation of transcription at cryptic promoter sites (Carvalho *et al.* , 2013). In addition, analysis of sequencing data from a SETD2 deficient cell line showed an in-

crease in intragenic initiation, which is the initiation of transcription at a cryptic promoter rather than the usual promoter site, in at least 11% of the active genes (Carvalho *et al.* , 2013). The aim of this work is to create an algorithm to detect intragenic initiation using count data from an RNAseq experiment with SETD2 deficient cells from six ccRCC cell lines.

1.5 Objectives

In the present work, we want to develop a method to identifiy genes with intragenic initiation using data from a RNA-seq experiment. The purpose of this is to help elucidate the function of the SETD2 gene using cell lines of ccRCC that contain this mutation and compare them to normal controls. The biological hypothesis behind this is that loss of function of the SETD2 gene will increase the amount of intragenic initiation. Our method will be based on transforming RNA-seq data into proportions and using two methods of comparison of proportions within each sample. After implementing these methods and obtaining results, we will compare them with the results obtained by Carvalho *et al.*, 2013.

Chapter 2

Statistical background

Statistical analysis of RNA-seq data aims to identify the genes, exons or transcripts that are differentially expressed between samples (Oshlack *et al.*, 2010). In addition, these methodologies permit identification of different patterns of expression (isoforms) between samples due to intragenic initiation and alternative splicing. There are several methods used in the analysis of differential expression of RNA-seq data which can be divided in parametric and non-parametric methods. Some have been incorporated into user-friendly packages that facilitate the analysis (e.g. edgeR (Robinson *et al.*, 2010), DEGseq (Wang *et al.*, 2010)).

Despite the fact that the methods of analysis of RNA-seq can detect differences in expression of a gene, exon or transcript individually, they are not designed to detect more complex phenomena like intragenic initiation and alternative splicing which depend on patterns of expression between each unit and a deep understanding of cell biology in order to model them.

In this chapter, we will present the basis for statistical hypothesis testing, a cornerstone in almost every statistical tool, the methods used in the statistical analysis of RNA-seq data and the background behind our proposed approach to detect intragenic initiation of transcription.

2.1 Statistical Inference

Statistical inference is the process of extracting conclusions from datasets containing observations from random variables. Since it is generally impossi-

ble to observe all the units/individuals in a population, statistical inference extrapolates conclusions using a representative sample of the population, given a certain degree of uncertainty.

One way of drawing conclusions from the data is by recurring to hypothesis testing. This approach implies simplifying the problem by dichotomizing it into two complementary hypotheses: the null hypothesis, H_0 and the alternative hypothesis, H_1. The null hypothesis is a claim contradictory to what we aim to prove and, naturally, the alternative hypothesis corresponds to the claim we believe is true and want to prove. The hypotheses are set as equalities or inequalities about parameters of the population like the expected value, the variance or the distribution of the population. For example, let X be the height of an individual in a population. If we want to show that the average height, $E(X) = \mu$, in a population is different from 170 cm, the hypotheses would be formulated like this: $H_0 : E(X) = 170 \ \ vs. \ \ H_1 : E(X) \neq 170$.

After establishing the hypotheses, we must make assumptions about the data, like the distribution of the random variable of interest and/or the independence of the observations, and calculate a value for the test statistic. The observed value of the test statistic is calculated from the data assuming that the null hypothesis is true. A statistically significant result corresponds to rejecting the null hypothesis in favor of the alternative hypothesis. To define if a result is statistically significant, we must determine a critical value and compare it to the value of the test statistic that will allows us to decide whether or not to reject H_0. If the observed test statistic exceeds the critical value, H_0 is rejected and we consider the alternative hypothesis H_1, which is usually what we want to prove.

Since we are inferring information about a population using only a sample, our conclusions may be incorrect due to chance. Therefore, when performing a statistical test we can commit two errors: reject the null hypothesis when it is true or fail to reject the null hypothesis when H_1 is true. These mistakes correspond to the type I and type II errors, respectively. In fact, the probability of type I error is strictly related with the critical value since the later corresponds to the $1 - \frac{\alpha}{2}$ quantile (in case of bilateral tests) of the distribution of the random variable being considered.

Statistical inference can be used to test hypotheses related with gene expression. The simplest question a molecular biologist can ask is what is the difference in expression of gene A between two cell types. In this regard, we can apply statistical hypotheses testing and formulate the following hypothe-

ses: H_0 : Gene A is equally expressed between the two cells types *vs.* H_1 : Gene A is differentially expressed between the two cell types. This is one possible application of statistics to the analysis of genetic data. However, many other applications exist, that span from gene expression, to protein structure and to epigenetics. Here we emphasize the applications of statistics in the analysis of NGS data.

2.2 Statistical analysis of differential expression of RNA-seq data

RNA-seq data is essentially count data of each experimental unit, whether it is a gene, exon or transcript. Therefore, the parametric methods used need to be able to model low count data of a small number of samples, which is generally the case with NGS experiments. The most common parametric methods used in RNA-seq data analysis are based on the Poisson and negative binomial distributions (Oshlack *et al.*, 2010). The binomial negative distribution can be seen as a generalization of the Poisson distribution (in which the mean equals the variance). In the binomial negative distribution the variance can be modeled separately which may be required in experiments with a small number of replicates to avoid type I errors (Anders and Huber, 2010). Several statistical analysis packages implement methodologies based in these two distributions: `edgeR`, `DEGseq`, `DESseq` (Anders and Huber, 2010), `DEXseq` (Anders *et al.*, 2012) and `bayseq` (Hardcastle *et al.*, 2010). All these packages are open source and available at Bioconductor (`http://www.bioconductor.org`).

Fisher's exact test has been used as a non-parametric method to detect differential expression (DE) (Wang *et al.*, 2008, Brooks *et al.*, 2010). It is used in the statistical analysis of 2x2 contingency tables and can be applied in RNA-seq data analysis to compare each exon between samples (Table 2.1).

In addition, to Fisher's exact test two other non-parametric methods used to analyze RNA-seq data are *cuffdiff* (Trapnell *et al.*, 2010) and `NOISeq` (Tarazona *et al.*, 2011). The first method in based on the cufflinks framework adapted to the detection of DE. *Cuffdiff* builds a table with the expected variance of each condition taking into account the amount of reads that aligned

Table 2.1: Applying Fisher's exact text to detect DE. Fisher's test is can be used to compare the levels of expression of the same exon between samples (control *vs.* mutated sample). Repeating this procedure for each pair of consecutive exons between samples allows for detection of intragenic initiation.

	Control	Sample
$Exon_i$	Read counts	Read counts
Total Mapped Reads	$TotalMappedReads_{control}$	$TotalMappedReads_{mutant}$

with an experimental unit across all the replicates. *Cuffdiff* then estimates how many reads originated from each experimental unit and queries the table for the variance of that unit. Additionally, it takes into account read mapping uncertainty as happens with reads aligning in splice site junctions. `NOISeq` estimates the "noise distribution" which takes into account the sample variability within the same condition before testing for DE between samples with different conditions (controls *vs.* mutant cell lines, for example).

Few methods used in the analysis of microarray data have been adapted to RNA-seq analysis. Microarray data are treated as continuous measurements and, consequently, are modeled by continuous distributions (e.g. normal distribution, gamma distribution) (Soneson *et al.*, 2013). On the other hand, NGS data are count data (sometimes low-count) and thus inherently follow a discrete probability distribution. Nevertheless, the `limma` (Smyth GK, 2004) method has been adapted through data transformation to RNA-seq data analysis and two `limma`-based methods exist: `voom(+limma)` (Smyth *et al.*, 2004 and Soneson *et al.*, 2013) and `vst(+limma)` (Smyth *et al.*, 2004 and Anders and Huber, 2010).

2.3 Detection of Intragenic Initiation

Fisher's exact test has been used to detect intragenic initiation in a SETD2-deficient ccRCC cell line (Carvalho *et al.*, 2013). By applying Fisher's test to assess DE between exons in each sample as previously exemplified, this method detects intragenic initiation by comparing the results for every pair of consecutive exons in each gene. So, a gene is said to have intragenic initiation if at least the first exon is not differentially expressed and the downstream exons show DE, meaning that transcription started after the

first exon. This method does not provide information regarding the genes that have intragenic initiation in the control (and whether they show the same behavior in the experimental sample). It only detects the genes that show differential intragenic initiation between samples.

Here we propose a different approach where we transform count data in proportions and consider the problem of comparing a series of proportions in a sample (and the associated problem of multiple statistical inferences). Moreover, we present the Marascuilo Procedure, a method of comparison of multiple proportions that corrects for the problem of testing multiple inferences in a sample.

2.3.1 Comparison of multiple proportions

The comparison of multiple proportions can be achieved by using a test to compare two proportions and repeating it for every pair of proportions. Let A_k be a random variable defined by the number of reads that align to gene k. We know that A_k follows a binominal distribution and, since the sample size is high, we can approximate its distribution to the standard normal distribution under the central limit theorem. The hypotheses tested for each exon of gene k are the following: $H_0 : p_1 = p_2 \;\; vs. \;\; H_1 : p_1 \neq p_2$, where p_1 is the number of reads that align with exon 1 and p_2 is the number reads that align with exon 2. The test statistic Z, under H_0, is given by:

$$Z = \frac{\hat{p}_1 - \hat{p}_2}{\sqrt{\frac{\hat{p}_1(1-\hat{p}_1)}{n_1} + \frac{\hat{p}_2(1-\hat{p}_2)}{n_2}}} \sim N(0,1)$$

For each test the probability of falsely rejecting the null hypothesis is α, but if we consider k independent tests, this probability will be larger and is given by $1 - (1-\alpha)^k$. Considering $\alpha = 0.05$, when we perform 100 independent tests there is a 0.994 probability that at least one test results in a wrong conclusion. This problem occurs frequently when conducting a study where several hypotheses are tested using one dataset. This is known as the problem of multiplicity or multiple testing and occurs whenever a set of statistical inferences are considered simultaneously. For example, if the expected value of the height of a population in 170 cm and we test the afore mentioned hypotheses about X ($H_0 : E(X) = 170 \;\; vs. \;\; H_1 : E(X) \neq 170$), at a significance level $\alpha = 0.05$, using 100 independent samples, our expected value of wrong rejections is 5. So, in 5 out of 100 tests, we would arrive to a false claim.

In our case, to detect intragenic initiation we incur in a problem of multiple testing since we want to compare all pairs of consecutive exons of each gene in a cell. Therefore, a method to correct our p-values so that we do not erroneously reject the null hypothesis is warranted. There are several approaches to p-value correction of which the most simple and frequently used models are the Family-Wise Error Rate (FWER) and the False Discovery Rate (FDR). A very commonly used method is the Benjamini-Hochberg procedure, which models the FDR. The FWER models are more conservative which is why we chose a the BH procedure since it allows for more hypothesis to be rejected and the detection of more genes with intragenic initiation.

The Benjamini-Hochberg procedure

The Benjamini-Hochberg (BH) procedure is based on the estimation of the FDR which is the fraction of erroneously rejected null hypotheses (also known as "false discoveries") over the number of tests performed.

Benjamini and Hochberg (1995) consider the following model: Let m be the number of hypotheses being tested, m_0 the number of true null hypotheses and R an observable random variable that corresponds to the number of hypotheses rejected (table 2.2). U, V, T and S are unobservable random variables, where U corresponds to the true negatives, V to the false positives or false discoveries (type I error), T to the false negatives (type II error) and S to the true positives or true discoveries.

Table 2.2: Modelling the false discovery rate To model the FDR, Benjamini and Hochberg (1995), consider that m is the number of hypothesis being tested and m_0 is the number of true null hypothesis. V and S are random variables corresponding to the number of falses positives and true positives, respectively. V and S are used to calculate the proportion of null hypothesis erroneously rejected, Q. The FDR is the expectation of Q.

	Considered non-significant	Considered significant	Total
True H_0	U	V	m_0
Non-true H_0	T	S	$m - m_0$
	$m - R$	R	m

The proportion of null hypotheses erroneously rejected, Q, is a random variable given by $Q = \frac{V}{V+S}$ and the FDR or Q_e is the expectation of Q:

$$FDR = Q_e = E(Q) = E(\tfrac{V}{V+S}) = E(\tfrac{V}{R}).$$

When $V + S = 0$, $Q = 0$ since no hypothesis can be falsely rejected.

The BH procedure controls the FDR at a predetermined significance level α and works as follows:

1. Consider m hypotheses tests $H_1, \ldots, H_m$ and their corresponding p-values $P_1, \ldots, P_m$.

2. Order the p-values from lowest to highest, $P_{(1)}, \ldots, P_{(m)}$

3. Find the largest k such that $P_{(k)} \leq \frac{k}{m}\alpha$

4. Reject all the hypotheses $H_i, i = 1, \ldots, k$ whose p-values $P_{(i)}$ satisfy the condition $P_{(i)} \leq P_{(k)}$.

This procedure is valid considering m independent hypotheses. Very frequently however, this assumption is not correct since there is a dependence structure between observations. In our case, we are comparing the expression among exons of the same gene and these observations are therefore correlated at the gene level. We need to consider this structure when correcting for multiplicity. To cope with problems like this some alterations have been introduced to the Benjamini-Hochberg procedure that consider a dependence structure between the hypotheses (Benjamini and Yekutieli, 2001; Hu *et al.*, 2010). In particular, Hu *et al.* (2010) developed the grouped Benjamini-Hochberg procedure which we will discuss.

False discovery rate control with groups

In multiple hypothesis testing, assuming independence between hypotheses may not translate the true structure of the data and may lead to erroneous results. A way adopted by several authors has been assigning weights to the hypotheses (or p-values) to create group structure (Efron, 2008; Genovese *et al.*, 2006 and Kang *et al.*, 2009). More recently, Hu and colleagues (2010) proposed the grouped Benjamini-Hochberg (GBH) where FDR control is achieved by weighing p-values.

The GBH procedure is an extension to the classic Benjamini-Hochberg (BH) procedure where the m hypotheses being tested can be individualized into K disjoint groups with sizes n_g, $g = 1, \ldots, K$.

In fact, when using the GBH procedure, we can consider two scenarios:

the oracle case, where the true proportion of null-hypotheses is known and the adaptive process, where this proportion is unknown and the proportion of true null hypotheses in each group is estimated by $\widehat{\pi_{g,0}}$. In the oracle case, the GBH procedure works as follows:

1. For each p-value in group g, calculate the weighted p-values, $P^w_{g,i} = \frac{\pi_{g,0}}{\pi_{g,1}} P_{g,i}$, where $\pi_{g,0}$ is the proportion of true null hypothesis in group g, $\pi_{g,1}$ is the proportion of true alternative hypothesis in group g and $P_{g,i}$ is the p-value associated with the i^{th} ordered element of each.

2. If $\pi_{g,0} = 1$, accept all the hypotheses for that group and $P^w_{g,i} = \infty$. If $\pi_{g,0} = 1$ for every g, then accept all null hypotheses for all groups and stop the procedure. Otherwise, perform the next steps.

3. Pool all the weighted p-values $P^w_{g,i}$ and order them from lowest to highest $P^w_{g,(1)} \leq ... \leq P^w_{g,(N)}$.

4. Find:

$$k = max\{i : P^w_{g,(i)} \leq \frac{i\alpha^w}{N}\}, \text{ where } \alpha^w = \frac{\alpha}{1-\pi_{g,0}}$$

5. If such a k exists, reject the k null hypotheses associated with $P^w_{g,(1)} \leq ... \leq P^w_{g,(k)}$. Otherwise, do not reject any of the hypotheses.

In practice, however, the true proportion of null hypotheses in each group is not known and it must be estimated ($\widehat{\pi_{g,0}}$). There are several methods used to estimate the proportion of null hypotheses of which the most emblematic are: the Least Slope (LSL) estimator (Benjamini and Hochberg, 2000), the Two-Stage (TST) method (Benjamini *et al.*, 2006) and Storey tail proportion of p-values method (Storey *et al.*, 2004). Here, we will only specify the LSL estimator which performs well in situations with a sparse signal, that is, with few rejected null hypotheses (Benjamini and Hochberg, 2000) which is our case.

The estimation of $\pi_{g,0}$ follows two steps (Hu *et al.* , 2010):

1. Compute $l_{g,i} = \frac{(n_g+1-i)}{1-P_{g,(i)}}$. As i increases, stop the first time the condition $l_{g,i} > l_{g,i-1}$ is met.

2. Calculate the LSL estimator for each group given by: $\gamma_g^{LSL} = min\left(\frac{\lfloor l_{g,i} \rfloor + 1}{n_g}, 1\right)$

The adaptive GBH can now be performed by replacing $\pi_{g,0}$ with γ_g^{LSL}. This procedure has been implemented in the R package `StructSSI` (Sankaran,

2010) which we will use in our analysis in the next chapter.

2.3.2 The Marascuilo procedure

The Marascuilo procedure is a statistical method that compares multiple proportions pair-wise and permits the identification of the pairs that are significantly different. The procedure incorporates the correction for multiple tests thus avoiding the need to use a p-value correction method.

The hypotheses tested in the Marascuilo procedure are: $H_0 : p_1 = p_2 = \ldots = p_m \quad vs. \quad H_1 : \exists i, j \; i \neq j : p_i \neq p_j$. Considering k samples, of sizes n_i $(i = 1, 2, ..., k)$, the first step is to calculate the test statistics which are given by the differences between each pair of proportions $p_i - p_j$, $i \neq j$. Then, critical values, r_{ij} are calculated for each pair from:

$$r_{ij} = \sqrt{\chi^2_{1-\alpha,k-1}} \sqrt{\frac{p_i(1-p_i)}{n_i} + \frac{p_j(1-p_j)}{n_j}} \tag{2.1}$$

The critical values are compared to the test statistics. If at least one absolute value of the test statistic $|p_i - p_j|$ exceeds the critical value the null hypothesis is rejected and we conclude there is a significant difference between samples i and j. By calculating r_{ij} using a χ^2 distribution with $k-1$ degrees of freedom, the Marascuilo procedure incorporates the correction for multiple tests. Here we use the Marascuilo procedure as statistical method to identify intragenic initiation. This approach is slightly different from most RNA-seq analysis methods that normally compare each experimental unit between samples. Both the Marascuilo Procedure and the test of comparison of multiple proportions with an associated p-value correction method will be applied to a set of RNA-seq data from ccRCC cells carrying mutations in the SETD2 gene with the purpose of elucidating the role of SETD2 in intragenic initiation.

Chapter 3

Method Description and Results

This chapter is a description of the implementation of the statistical methods described in the previous chapter and the results obtained after applying these methods. It starts with a description of the data and data processing performed followed by a detailed description of the data. We then present a detailed description of how we applied each method - the two proportions comparison test and the Marascuilo procedure - and the results obtained.

The steps described here translate the contents of the scripts that are attached to this manuscript (Appendix A).

3.1 Data Description and Processing

The data used in our analysis was RNA-seq data produced by Sérgio de Almeida Lab at Instituto de Medicina Molecular (IMM). Six ccRCC cell lines, 2 SETD2 wild-type cell lines (Caki1 and Caki2) and 4 lines carrying knock-out mutations in the SETD2 gene (AB, ER, FG2 and MF) were analyzed. As previously mentioned, sequencing data was obtained with the Illumina® Genome Analyzer HiSeq2000 (Bentley *et al.*, 2008).

The procedure for data analysis followed the workflow depicted in Figure 1.6. We were granted access to raw data in .fastqc format and used TopHat version 2.0.9 (Trapnell *et al.*, 2012) to map high-throughput sequencing reads to the reference human genome (hg19) (Deszer *et al.*, 2012). Reads mapping

to multiple locations were excluded. We used BEDTools version 2.17.0 (Quinlan and Hall, 2010) to obtain count data for each exon. The data obtained was analyzed using R version 3.0.2 for Mac OS X.

The steps of data transformation and analysis were as follows (Figure 3.1):

1. **Identification and exclusion of genes with 4 or less exons**. Considering genes that have few exons will increase the number of false positives in our approach. Intragenic initiation relies upon assuming that at least the first exon is not expressed, that transcription starts in a downstream exon and expression is stable after that exon. We opted to exclude genes with few exons in order to avoid misclassification of genes due to exon number. A gene with few exons is more likely to obey to the criteria we used that includes a maintenance of higher expression of the downstream exons after intragenic initiation (ITI).

2. **Normalization**. We normalized the data by converting reads counts to *reads per kilobase of transcript per million reads mapped* (RPKM) as described by Mortazavi *et al.*, 2008. To nornalize read counts we used the following formula:

$$RPKM = \frac{10^9 \cdot Number\,of\,mapped\,reads}{Total\,mapped\,reads \cdot Exon\,length\,in\,kilobase{-}pairs}$$

This step is essential to ensure that expression levels between exons are comparable since without normalization, read counts are proportional to the number of base pairs in an exon and total number of reads mapped for each sample.

3. **Selection of transcriptionally active genes**. We excluded the bottom third (first tercile) of genes with less expression as genes that were not transcriptionally active . Genes on the upper third (third tercile) of expression were considered to be "very active", that is, highly expressed. Expression of a gene was calculated as the sum of normalized read counts of the exons of a gene.

4. **Estimating reads counts based on the more active genes**. To calculate proportions of gene expression, we calculated the expected number of reads for each exon assuming that all the genes are equally active and in the "very active" group. The estimated read counts were normalized, in order to obtain one single value for the estimated read counts across all exons.

5. **Calculating proportions of gene expression**. The estimated read

counts were used (denominator) to calculate the normalized proportion of expression of each exon. Some of the genes for each sample showed proportions that were greater than 1 thus making them not amenable to be analyzed with our proposed methods. The few genes in this group were transfered to a new matrix and were analyzed separately.

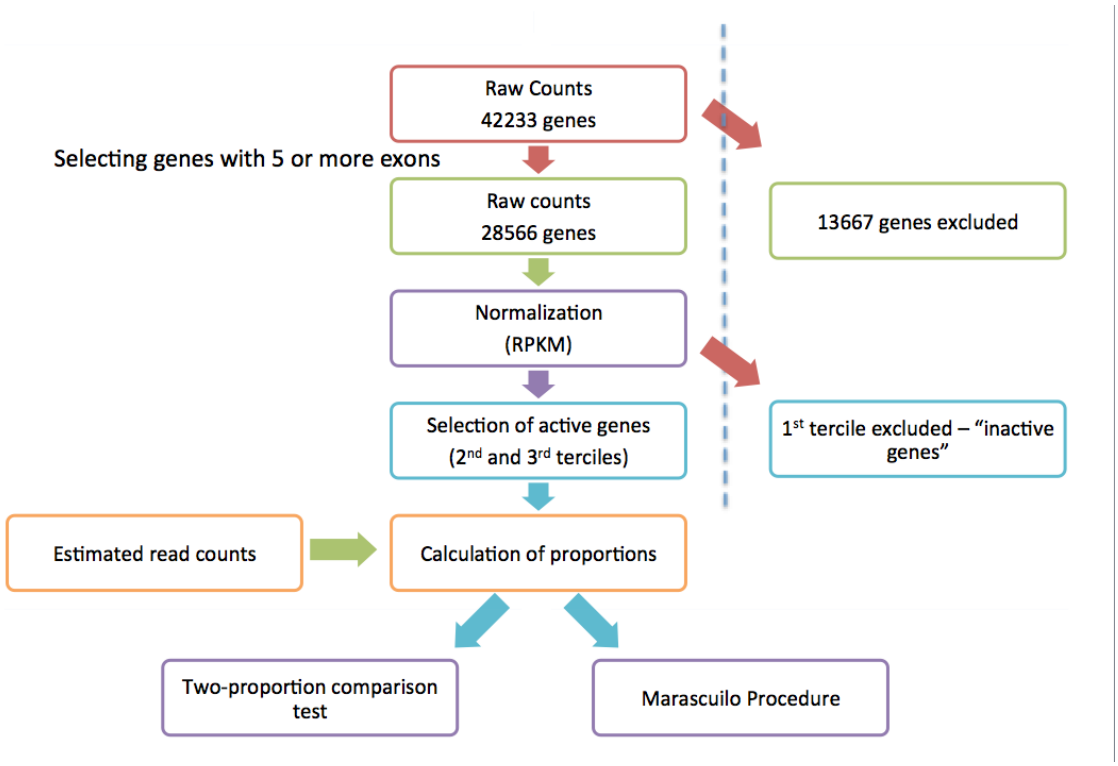

Figure 3.1: Data processing workflow.

After applying these steps, we obtained a vector with proportions of expression of each exon for each of our six samples. The two proportion comparison test and the Marascuilo procedure, described below, used these as the input for data analysis.

3.2 Samples and data processing

After processing the data using TopHat and BEDTools, two commonly used software tools in NGS data analysis, we transferred the data to R and built the data matrix represented in Table 3.1.

Table 3.1: Raw Data Matrix The data matrix with non-normalized counts was obtained by importing to R the .bed file generated by BEDTools. This file also contains data with a gene tag, chromosome and exon number for each gene. This data was parsed to columns in our matrix. Additionally, we matched the gene name to each tag based on the human genome (hg19) annotation available at UCSC Genome Browser (Kent *et al.*, 2002) and added an index in order to perform our analysis by gene.

gene	exon	chromossome	caki1	caki2	rccab	rccer	rccmf	rccfg2	gene name	index
NM_032291	0	chr1	86	67	7	147	1	0	SGIP1	1
NM_032291	1	chr1	52	38	5	104	2	0	SGIP1	1
NM_032291	2	chr1	37	17	1	82	0	0	SGIP1	1
...	...	...	...	...	...	...	...	...	...	...
NM_012425	0	chr10	28970	26891	20143	6718	9797	12766	RSU1	21101
NM_012425	1	chr10	6625	5979	3737	1532	2162	3060	RSU1	21101
NM_012425	2	chr10	6681	5933	3468	1499	2053	2977	RSU1	21101
NM_012425	3	chr10	5894	5370	3152	1296	1842	2674	RSU1	21101
...	...	...	...	...	...	...	...	...	...	...

Following our data processing algorithm, we identified 42233 genes, excluding isoforms, of which 13667 genes (≈31.7%) were excluded because they were composed of less than 5 exons.

The raw counts of the remaining 28566 genes were normalized and were used to calculate proportions of expression for each exon. In all samples, we identified a small subset of very active genes that had at least one exon with proportions greater than one. The number of genes in each sample that belong to this subgroup is represented in Table 3.2. The differences are due to the different patterns of expression across samples but in all cases this corresponds to a small subset of genes.

Table 3.2: Genes with proportions greater than 1 For each sample the number of genes with proportions greater than 1 is displayed. These correspond to a very small subset of genes comparing to the number of genes that was analyzed. For each sample, the genes with a pattern suggestive of ITI were selected.

Sample	Genes with proportions >1	Genes with possible ITI
Caki1	10	'SPP1', 'AKR1B1', 'VIM', 'GAPDH', 'RPL13'
Caki2	7	'VIM', 'GAPDH', 'RPL13'
AB	7	'SPP1', 'RPL17'
ER	14	'ENO1', 'SPP1', 'NPM1', 'ALDOA', 'RPL17'
MF	8	'HLA-B', 'VIM', 'PSAP', 'RPL17'
FG2	12	'ENO1', 'SPP1', 'TGFBI', 'VIM', 'RPS24', 'GAPDH'

These genes were eliminated from the analysis of proportions because the methods implemented can only handle proportions in the interval $[0, 1]$.

Curiously, some of these genes showed a pattern suggestive of intragenic initiation, where the proportion of a downstream exon was higher than the expression of the upstream exon/s (Table 3.3).

Since these genes are a small subset of expression outliers, we performed an analysis "by hand" to detect the genes that had a pattern suggestive of intragenic initiation. The criteria for selection of these genes was:

(1) Existence of several proportions > 1 (more than half of the number of exons) and the first exon/s with lower expression

or

(2) Few proportions > 1 (less than half of the number of exons), several proportions with levels within 0.25 of the proportions > 1 and the first exon/s with lower expression.

The sum of the number exons with proportions > 1 and the number of exons within 0.25 these had to correspond to at least half the number of exons in the gene. The gene GAPDH is an example of the application of criterion (1). After identifying and isolating these genes, the estimated read counts for the remaining genes were recalculated in order to translate the expected read counts unbiased by the genes with proportions greater than one.

Table 3.3: Expression pattern of two genes with ITI and proportions greater than 1.

Gene ID	Gene	Expression proportion
NM_001256799	GAPDH	0.000463
NM_001256799	GAPDH	1.468526
NM_001256799	GAPDH	1.541777
NM_001256799	GAPDH	1.434993
NM_001256799	GAPDH	1.542627
NM_001256799	GAPDH	1.904736
NM_001256799	GAPDH	1.129369
NM_001256799	GAPDH	0.780231

3.3 Detection of intragenic initiation

3.3.1 Two proportions comparison test with the grouped Benjamini-Hochberg procedure

The two proportions comparison test was applied considering each exon as a unit grouped at the gene level. We compared all pairs of contiguous exons of each gene (for a gene with j exons, $j-1$ comparisons were made) and obtained a set of p-values uncorrected for multiple testing. For gene i with j exons, our null-hypothesis was $H_0 : p_{i,1} = p_{i,2} = \ldots = p_{i,j}$.

After calculating a p-value for each pair-wise comparison, we applied the grouped Benjamini-Hochberg procedure. As previously mentioned, our group structure was defined at the gene level. However, a large number of groups as occurs with our data leads to non-rejection of H_0 in all cases and all the weighted p-values are set to be infinite. This occurs because the GBH procedure assigns a value of "-Inf" (where "Inf" stands for infinity) when the proportion of true null hypothesis in a group is 1. Given our very large number of genes (or groups), the estimated proportion of H_0 is 1 thus not allowing the use of this p-value correction method.

Due to this and similarly to Carvalho *et al.*, 2013, we assumed no hierarchical structure and the standard BH procedure was applied. After calculating the p-values and establishing significance assuming a type I error probability of 0.05, we discarded all the genes that did not contain any pair of exons with significant differential expression. We then selected only the genes that showed higher expression in the downstream exon of the first pair of exons that showed differential expression (DE), thus eliminating genes whose first exon had more expression than the second, for example). The first pair with differential expression had to be in the beginning of the gene to avoid selecting genes with DE in one of the last exons (thus not corresponding to genes with ITI). For this, we selected only the genes whose first pair of exons with differential DE was located in the first 40% of exons of that gene. Moreover, we selected only the genes with stable expression levels in the downstream exons. For this, we assumed that no more than 50% of the p-values detected by our method were below the 0.05 level of significance.

After gene selection, we matched the gene tags to gene names and obtained lists of the identified genes with ITI for each sample.

Results

Using the two proportion comparison test repeated over every pair of consecutive exons in a gene along with the Benjamini-Hochberg procedure for p-value correction, we identified very few genes (<12) with suspected ITI (Table 3.4). These correspond to a total of 20 uniquely identified genes from the six samples.

Table 3.4: Genes with intragenic transcription initiation using the two proportions comparison test. The first column indicates the number of genes detected for each sample. The second column corresponds to the genes identified with the same method without making any restrictions regarding the expression levels of the downstream exons.

.

Sample	Number of identified genes	Number of detected genes without downstream exon expression restrictions
Caki1	9	214
Caki2	5	189
AB	3	90
ER	2	53
MF	9	279
FG2	11	346

Since our criteria to include a gene in the set of genes with ITI was relatively stringent regarding the expression of the exons downstream of the first expressed exon, we considered all the genes detected regardless of the expression levels of the downstream exons (Table 3.4). The number of genes represented in the second column of the table corresponds to the maximum number of genes that can potentially have ITI as detected by this method. However, some of these genes may not correspond to genes with ITI but to alternative spliced forms instead.

Whitin this larger group, 382 unique genes can be identified. This is a very small number as compared to the results of Grosso *et al.*, 2013 where 1037 genes were identified after applying Fisher's exact test as described in Carvalho *et al.*, 2013. The concordance between this method and the implementation of the two proportion comparison test is negligible with ≤ 2 genes being concordant in several comparisons. We hypothesized that this difference could have occured because the data was transformed into proportions. Therefore, we compared these results with those obtained when we apply the Maracuilo procedure (described below). Once again, the concordance was negligible.

3.3.2 The Marascuilo procedure

Implementing the Marascuilo procedure

Analogously to the two proportion comparison test, we applied the Marascuilo procedure to compare pairs of exons in every gene. This procedure makes every pair-wise comparison of exons in each gene as opposed to the previous procedure where the number of comparisons is the `number of exons`-1. For a gene with j exons, ${}^{j}C_{2}$ comparisons are performed and a p-value is obtained for each. Our goal, however, is to consider only the comparisons between contiguous exons and, therefore, selection of the relevant pair-wise comparisons was made. The Marascuilo procedure incorporates correction for multiplicity thus not requiring the use of the Benjamini-Hochberg procedure or other method of p-value correction.

Selecting only the comparisons of interest can be performed because it does not bias the results towards the alternative hypothesis. In fact, performing multiple testing p-value correction for more pairs of proportions than we are interested in, increases the p-values thus biasing the conclusions towards H_0 (more conservative test).

Like in the previous method, we selected the genes that showed significant higher expression in a downstream exon (considering $\alpha = 0.05$) and defined that no more than 50% of the p-values in the downstream comparisons analyzed were below the 0.05 level of significance. After selecting the pairs of comparisons of interest we filtered our results to the two proportion comparison test and matched the gene tags to gene names and obtained lists of the identified genes with ITI for each sample.

Results

Using the Marascuilo procedure, we identified, approximately, 500 genes per sample with suspected ITI (Table 3.5):

Table 3.5: Genes with intragenic transcription initiation using the Marascuilo procedure. The first column corresponds to the number of genes detected for each sample. The second column corresponds to the genes identified for each sample and that are not contained in the Caki1 (control) sample.

Sample	Marascuilo Procedure - Number of identified genes	Number of genes detected excluding intersection with Caki1
Caki1	514	-
Caki2	517	208
AB	541	287
ER	490	299
MF	492	298
FG2	526	259

These counts correspond to a total of 1304 different genes identified since there is considerable gene overlap between samples. This overlap is illustrated in the Venn Diagrams in Figure 3.2. The diagrams show that even though there is a similar number of genes detected for each sample, there are more genes detected in the mutated cell lines: first, when we exclude from Caki2 and the four mutant lines the genes that are also detected in Caki1, we observe that each of the mutated cell lines has near 300 genes with intragenic initiation as opposed to Caki2 that shows only 208 genes (Table 3.5); second, when we analyze Figure 3.2 b, we observe that there are 583 different genes in the 4 mutant cell lines which corresponds to an average of 146 separate genes per sample. In the controls there are 223 genes detected that are not detected in the mutant cell line which corresponds to an average of 112 genes per sample. Even though this is a small difference, it suggests that the SETD2 mutant cell lines have, on average, more ITI than the controls. Note that in table 3.5 we consider Caki1 as baseline and compare the differences between the other five cell lines after extracting the genes these have in common with Caki1. This is meant to make our results comparable to those obtained by the method described in Carvalho *et al.*, 2013.

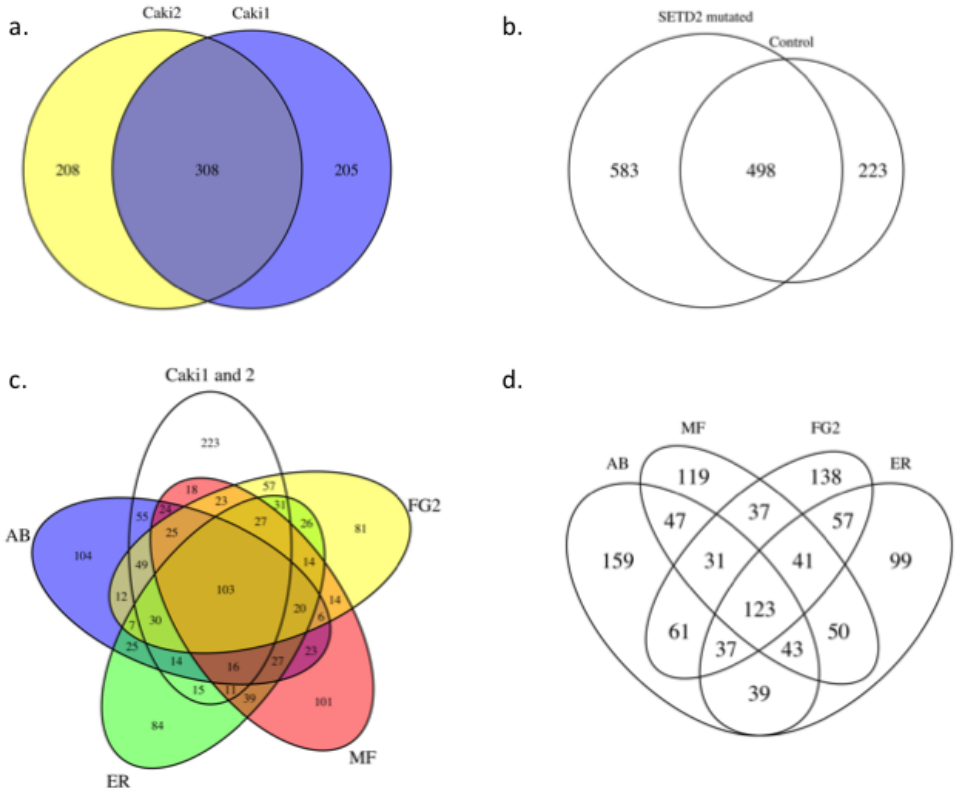

Figure 3.2: Venn diagrams of genes with Intragenic initiation.
a. Genes identified in the 2 control groups: Caki1 and Caki2. **b.** Genes identified in the set of control cell lines vs. the set of all genes identified in a mutant cell line. **c.** Gene overlap between the controls and the 4 mutant cell lines. **d.** Gene overlap between the four mutant cell lines.

Our results also show that ITI most often starts in the second exon (77.7% of the identified genes in the mutated cell lines) as is shown in Figure 3.3. This is in agreement with the results by Grosso *et al.*, 2013 where the second exon was also the most common site on intragenic transcription initiation (36% of the detected genes). This difference is attributable to the constraints we added for the first exon where transcription started to be in the first 40% of exons in each gene.

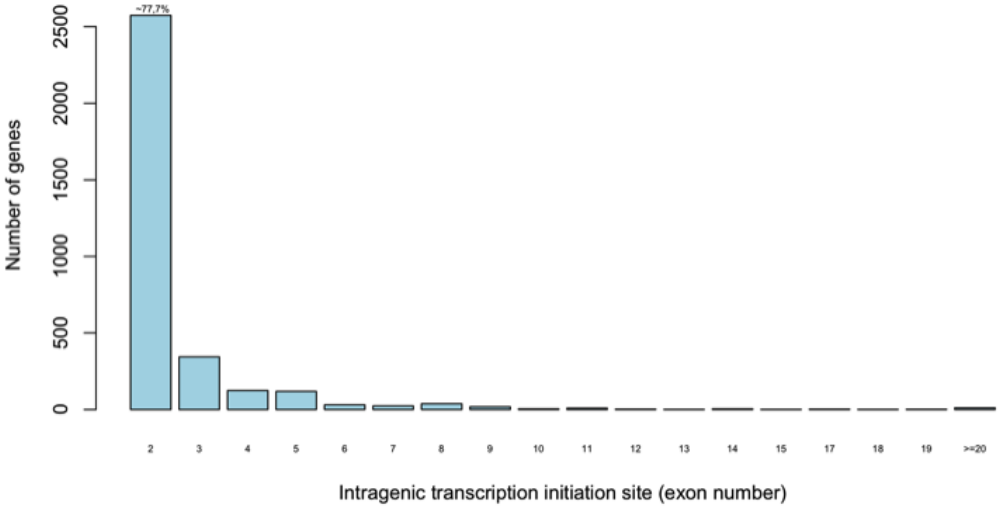

Figure 3.3: Bar plot of intragenic initiation site location. The x-axis represents the exon at which transcription starts. Approximately 78% of genes with ITI start transcription at the second exon.

We had access to the list of genes with ITI detected by Grosso *et al.*, 2013 and compared our results to the one obtained by their method (Carvalho *et al.*, 2013). Overall, we detected a similar number of genes with ITI: 1304 genes *vs.* 1037 genes detected by Grosso *et al.*, 2013. If we exclude the genes identified by our method in the controls Caki1 and Caki2 (which are not detected using Fisher's exact test), we identify 583 genes (Figure 3.2 b). This is in agreement with our predictions since the Marascuilo procedure is expectedly more conservative than Fisher's exact test because more hypothesis are tested thus leading to a smaller number of detected genes.

Moreover, we assesed the concordance between the mutant cell lines between the two methods. We verify that the concordance is relativelly small. For the samples AB, ER, MF and FG2 the concordance between methods is 3.0%, 6.3%, 4.5% and 12.7%, respectively (Figure 3.4). On the contrary, when we compare the intra-SETD2 mutated cell line concordance (Figure 3.2 d) with the similar diagram built by Grosso *et al.*, 2013, we verify that our mutated cell lines have 123 concordant genes between the four samples (*vs.* 9 genes in Grosso *et al.*, 2013). The same is apparent with the 2 and 3 mutant cell line concordance. Grosso *et al.* detected 1037 genes with 22.3% of genes shared among at least 2 of the 4 SETD2 deficient cell lines. In our case, the same concordance was in the order 52.3% which suggest that our method generates more consistent results.

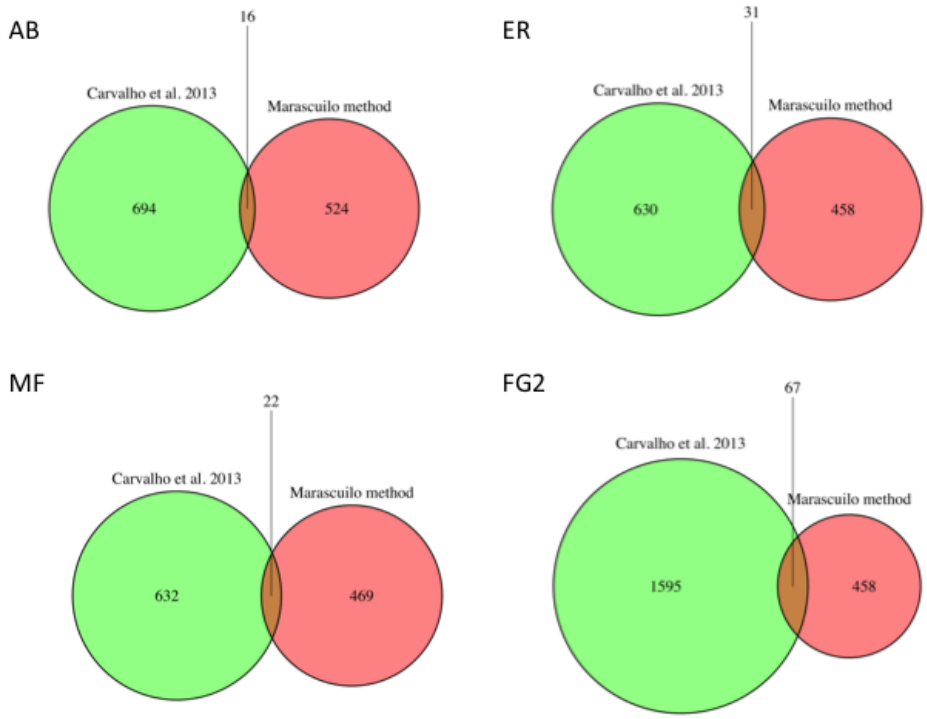

Figure 3.4: Venn diagrams - Marascuilo procedure-based method *vs.* **Carvalho** *et al.* **method.** Venn diagrams of the intersection between the Marascuilo procedure-based method and the method described by Carvalho *et al.*, 2013.

Chapter 4

Discussion

In this manuscript, we applied two statistical methods to detect intragenic initiation using high-throughput sequencing data. Both methods were based on transforming count data into proportions and applying tests that allowed comparison of multiple proportions.

In this work, an emphasis was put on the statistical methodology and the initial processing of RNA-seq was performed in a relatively standard and straightforward way. It should, however, be acknowledged that there are also several methods of processing NGS data and that this is a very active research field. Every step in the NGS data analysis workflow, from mapping to alignment, summarization and normalization is undergoing improvements with the development of new and more accurate tools for data processing. These are meant to improve the quality of the data and reflect on the quality of the conclusions that can be derived from it.

Regarding the statistical methodology, the main focus of this work was to create a method that would detect the phenomenon of intragenic initiation by comparing the read counts obtained in four samples of ccRCC SETD2 knock-down cell lines and compare them with 2 normal controls. Even though NGS data is count data, we transformed our variables and obtained proportions based on the normalized read counts and the expected reads counts for each gene assuming that every each is equally active. However, using proportions in this way carries the problem that some genes that are expression outliers have at least one pair of exons with proportions greater than 1 which does not allow these genes to be included in the analysis. Nevertheless, this is not a major limitation of the method since the the number of genes in this group corresponds to less than 0.05% of the analyzed genes.

The main difference between our approaches and the one applied by Carvalho *et al.*, 2013 was that we did not use direct comparison between mutant cell lines and controls to detect intragenic initiation, that is, controls are not used as a baseline and the method identifies the differences between controls and mutant cell lines. This is relevant because: (1) intragenic initiation may already occur in cells with a wild-type SETD2 gene and (2) the SETD2 gene may be a driver of ITI or it may change the genes pattern that has intragenic initiation. By using controls as baseline, the difference suggested in (2) cannot be assessed. In contrast, using exon expression comparisons within each gene to identify ITI permits identification of which genes in a sample show a pattern of ITI. In this fashion, it is possible to define which genes have ITI in the controls and then compare with the mutant cell lines. In fact, the results obtained are more suggestive of (2) as opposed to (1) since the number of genes detected using the Marascuilo procedure was similar across the six samples. However, when we compare Caki2 and the four mutant cell lines with Caki1 as is performed in Carvalho *et al.*, 2013 and Grosso *et al.*, 2013, we verify that there are more genes with intragenic initiation in the mutant samples which is suggestive of (1). We attribute the smaller difference than the one encountered by Grosso *et al.*, 2013 to the conservativeness of the Marascuilo procedure.

The two proportion comparison method approach yielded very few genes as compared to the Marascuilo procedure even though the later is a more conservative test. We hypothesized that this significant difference was due to the detection of small fluctuations in exon expression by this method. In fact, if we relax the assumptions do define ITI, we detect more genes (Table 3.4). However, the absence of intersection of these genes with the results obtained with the Marascuilo procedure and by Grosso *et al.*, 2013 is remarkable. A possible explanation for this is the detection of excessive noise in each gene that interferes with the detection of the genes with ITI.

When we applied the two proportions comparison test, we first used the grouped Benjamini-Hochberg procedure to respect the group structure in the data: exons-genes-genome. Given the large number of groups (approximately $20,000$, corresponding to the number of genes analyzed), we were not able to detect any differences in expression since we could never reject H_0 in any of the comparisons. For this reason, we applied the Benjamini-Hochberg procedure and assumed independence between genes in each sample which is similar to the procedure applied by Carvalho *et al.*, 2013. This does not explain the small number of genes detected since not considering group structure is less conservative than considering a group structure. Therefore, this

procedure should detect more genes as opposed to the application of the grouped Benjamini-Hochberg procedure.

The Marascuilo procedure-based methodology seems to be able to detect genes with intragenic initiation even though there was a weak concordance between our results and results previously described (Grosso *et al.*, 2013). The differences encountered are attributable to the selection criteria that each method uses. Particularly, in our case we are stringent about the position of the first differentially expressed exon and the number of differentially expressed exons downstream of the first expressed exon.

The restrictions imposed on the number of differentially expressed consecutive exons downstream of the first spot where DE was detected are of particular importance. In particular, when we consider the method of Carvalho *et al.*, 2013, the restriction is put on 60% of downstream exons having differential expression comparing to the control. In our approach, our main goal is to identify genes that have only one pair of differentially expressed exons. In short, the previously described method aims to identify genes with a high proportion of true alternative-hypotheses whereas we aim to identify genes with a high proportion of true-null hypotheses. We hypothesize that this is the main reason for the differences in our results. Additionally, data processing of the data was performed differently between our methods and Carvalho *et al.*, in particular regarding criteria for gene exclusion. In our approach a gene was considered active if the reads counts were above the first tercile whereas in Carvalho *et al.* exclusion is based on a fixed read count threshold.

While the method we applied based on the Marascuilo procedure performs p-value correction for multiple testing, it does not take into account the hierarchical group structure of exon-gene-genome. It corrects p-values for each pair-wise comparison at the gene level but assumes no group structure at the gene level. In addition, the Maracuilo procedure performs all possible pair-wise comparisons between the exons of each gene instead of the comparing only the consecutive exons. This biases our conclusions towards the non-rejection of the hypothesis that a gene has ITI, decreases the amount of genes detected by the method and, possibly, the amount of genes that overlap with the results obtained by Grosso *et al.*, 2013.

The use of RNA-seq data to detect phenomena like ITI can be used in two ways: (1) to evaluate the impact of genetic alterations in the pattern of ITT, which is the goal in our case or (2) as a survey method to identify specific genes that have intragenic initiation and that required further exploration.

In the first, ITI can be studied either in one sample or to compare different samples. In fact, we propose the experiment of comparing several sequencing data from the same cell line and evaluate the variation in results using the same method in order to be able to estimate the variability of the data. Only with this estimate it is possible to accurately evaluate if the differences encountered between methods are to be expected or not.

In summary, this work had the main goal of creating a procedure to identify ITI in SETD2 deficient cells. Nevertheless, this can be generalizable to detect intragenic initiation using RNA-seq data regardless of the gene whose function is to be elucidated. Given that our approach encompasses detection of ITI within a sample without comparing against a baseline, it can be . We propose the use of the Marascuilo procedure as a method to detect ITI in other settings and suggest the creation of more specific criteria to define that a gene had ITI in order to more accurately detect this phenomenon.

References

Anders, Simon, & Huber, Wolfgang. 2010. Differential expression analysis for sequence count data. *Genome biology*, 11(10), R106.

Anders, Simon, Reyes, Alejandro, & Huber, Wolfgang. 2012. Detecting differential usage of exons from RNA-seq data. *Genome Research*, 22(10), 2008–17.

Benjamini, Y, & Hochberg, Y. 2000. On the Adaptive Control of the False Discovery Rate in Multiple Testing With Independent Statistics. *Journal of Educational and Behavioral Statistics*, 25(1), 60–83.

Benjamini, Y, Krieger, AM, & Yekutieli, D. 2006. Adaptive linear step-up procedures that control the false discovery rate. *Biometrika*, 93(3), 491–507.

Benjamini, Yoav, & Hochberg, Yosef. 1995. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B*, 57(1), 289–300.

Bentley, David R, Balasubramanian, Shankar, & Swerdlow, et al. 2008. Accurate whole human genome sequencing using reversible terminator chemistry. *Nature*, 456(7218), 53–9.

Brooks, Angela N, Yang, Li, Duff, Michael O, Hansen, Kasper D, Park, Jung W, Dudoit, Sandrine, Brenner, Steven E, & Graveley, Brenton R. 2011. Conservation of an RNA regulatory map between Drosophila and mammals. *Genome research*, 21(2), 193–202.

Carvalho, Sílvia, Raposo, Ana Cláudia, Martins, Filipa Batalha, Grosso, Ana Rita, Sridhara, Sreerama Chaitanya, Rino, José,

Carmo-Fonseca, Maria, & de Almeida, Sérgio Fernandes. 2013. Histone methyltransferase SETD2 coordinates FACT recruitment with nucleosome dynamics during transcription. *Nucleic acids research*, 41(5), 2881–93.

Cooper, Geoffrey M., & Hausman, Robert E. 2007. *The Cell A Molecular Approach, 4th Ed. + Lecture Notebook*. Sinauer Associates Incorporated.

Dillies, Marie-Agnès, Rau, Andrea, Aubert, Julie, Hennequet-Antier, Christelle, Jeanmougin, Marine, Servant, Nicolas, Keime, Céline, Marot, Guillemette, Castel, David, Estelle, Jordi, Guernec, Gregory, Jagla, Bernd, Jouneau, Luc, Laloë, Denis, Le Gall, Caroline, Schaëffer, Brigitte, Le Crom, Stéphane, Guedj, Mickaël, & Jaffrézic, Florence. 2013. A comprehensive evaluation of normalization methods for Illumina high-throughput RNA sequencing data analysis. *Briefings in bioinformatics*, 14(6), 671–83.

Dreszer, Timothy R, Karolchik, Donna, Zweig, Ann S, Hinrichs, Angie S, Raney, Brian J, Kuhn, Robert M, Meyer, Laurence R, Wong, Mathew, Sloan, Cricket A, Rosenbloom, Kate R, Roe, Greg, Rhead, Brooke, Pohl, Andy, Malladi, Venkat S, Li, Chin H, Learned, Katrina, Kirkup, Vanessa, Hsu, Fan, Harte, Rachel A, Guruvadoo, Luvina, Goldman, Mary, Giardine, Belinda M, Fujita, Pauline A, Diekhans, Mark, Cline, Melissa S, Clawson, Hiram, Barber, Galt P, Haussler, David, & James Kent, W. 2012. The UCSC Genome Browser database: extensions and updates 2011. *Nucleic acids research*, 40(Jan.), D918–23.

Duns, Gerben, van den Berg, Eva, van Duivenbode, Inge, Osinga, Jan, Hollema, Harry, Hofstra, Robert M W, & Kok, Klaas. 2010. Histone methyltransferase gene SETD2 is a novel tumor suppressor gene in clear cell renal cell carcinoma. *Cancer research*, 70(11), 4287–91.

Efron, Bradley. 2008. Microarrays, Empirical Bayes and the Two-Groups Model. *Statistical Science*, 23(1), 1–22.

Fleischmann, R D, Adams, M D, White, O, Clayton, R A, Kirkness, E F, Kerlavage, A R, Bult, C J, Tomb, J F, Dougherty,

B A, & Merrick, J M. 1995. Whole-genome random sequencing and assembly of Haemophilus influenzae Rd. *Science (New York, N.Y.)*, 269(5223), 496–512.

Genovese, C. R., Roeder, K., & Wasserman, L. 2006. False discovery control with p-value weighting. *Biometrika*, 93(3), 509–524.

Gilbert, W, & Maxam, A. 1973. The nucleotide sequence of the lac operator. *Proceedings of the National Academy of Sciences of the United States of America*, 70(12), 3581–4.

Goeman, Jelle J, & Solari, Aldo. 2014. Tutorial in biostatistics: multiple hypothesis testing in genomics. *Statistics in medicine*, 33(11), 1946–1978.

Grosso, AR, Martins, F, Carvalho, S, Leite, AP, Desterro, J, Carmo-Fonseca, M, & Almeida, SF. 2013. *Epigenetic Misregulation and Transcriptome Alterations in Renal Carcinoma.*

Hu, James X, Zhao, Hongyu, & Zhou, Harrison H. 2010. False Discovery Rate Control With Groups. *Journal of the American Statistical Association*, 105(491), 1215–1227.

Jemal, Ahmedin, Siegel, Rebecca, Xu, Jiaquan, & Ward, Elizabeth. 2010. Cancer statistics, 2010. *CA: a cancer journal for clinicians*, 60(5), 277–300.

Kang, Guolian, Ye, Keying, Liu, Nianjun, Allison, David B, & Gao, Guimin. 2009. Weighted multiple hypothesis testing procedures. *Statistical applications in genetics and molecular biology*, 8(Jan.), Article23.

Kent, W. J., Sugnet, C. W., Furey, T. S., Roskin, K. M., Pringle, T. H., Zahler, A. M., & Haussler, a. D. 2002. The Human Genome Browser at UCSC. *Genome Research*, 12(6), 996–1006.

Kim, Jung, & Yu, Jindan. 2012. Interrogating genomic and epigenomic data to understand prostate cancer. *Biochimica et biophysica acta*, 1825(2), 186–96.

Luco, Reini F, Pan, Qun, Tominaga, Kaoru, Blencowe, Benjamin J, Pereira-Smith, Olivia M, & Misteli, Tom. 2010. Regulation of alternative splicing by histone modifications. *Science (New York, N.Y.)*, 327(5968), 996–1000.

Maradeo, Marie E, & Dulaimi, Essel. 2013. Aberrant promoter hypermethylation chromatin-modifying genes is absent or rare in clear cell RCC. 8(5), 486–493.

Marascuilo, LA. 1966. Large-sample multiple comparisons. *Psychological Bulletin*, 65(5), 280–290.

Marguerat, S, Wilhelm, BT, & Bähler, J. 2008. Next-generation sequencing: applications beyond genomes. *Biochemical Society transactions*, 36(5), 1091–6.

Maxam, A M, & Gilbert, W. 1977. A new method for sequencing DNA. *Proceedings of the National Academy of Sciences of the United States of America*, 74(2), 560–4.

Morozova, Olena, & Marra, Marco a. 2008. Applications of next-generation sequencing technologies in functional genomics. *Genomics*, 92(5), 255–64.

Mortazavi, Ali, Williams, Brian A, McCue, Kenneth, Schaeffer, Lorian, & Wold, Barbara. 2008. Mapping and quantifying mammalian transcriptomes by RNA-Seq. *Nature methods*, 5(7), 621–8.

Nagaraj, Shivashankar H, Gasser, Robin B, & Ranganathan, Shoba. 2007. A hitchhiker's guide to expressed sequence tag (EST) analysis. *Briefings in bioinformatics*, 8(1), 6–21.

Oshlack, Alicia, Robinson, Mark D, & Young, Matthew D. 2010. From RNA-seq reads to differential expression results. *Genome biology*, 11(12), 220.

Ozsolak, Fatih, & Milos, Patrice M. 2011. RNA sequencing: advances, challenges and opportunities. *Nature reviews. Genetics*, 12(2), 87–98.

Pan, Qun, Shai, Ofer, Lee, Leo J, Frey, Brendan J, & Blencowe, Benjamin J. 2008. Deep surveying of alternative splicing complexity in the human transcriptome by high-throughput sequencing. *Nature genetics*, 40(12), 1413–5.

Pattenden, Samantha G, Gogol, Madelaine M, & Workman, Jerry L. 2010. Features of cryptic promoters and their varied reliance on bromodomain-containing factors. *PloS one*, 5(9), e12927.

Quinlan, Aaron R, & Hall, Ira M. 2010. BEDTools: a flexible suite of utilities for comparing genomic features. *Bioinformatics (Oxford, England)*, 26(6), 841–2.

Risso, Davide, Schwartz, Katja, Sherlock, Gavin, & Dudoit, Sandrine. 2011. GC-content normalization for RNA-Seq data. *BMC bioinformatics*, 12(1), 480.

Robinson, Mark D, & Oshlack, Alicia. 2010. A scaling normalization method for differential expression analysis of RNA-seq data. *Genome Biology*, 11, R25.

Robinson, Mark D, McCarthy, Davis J, & Smyth, Gordon K. 2010. edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics (Oxford, England)*, 26(1), 139–40.

Saccone, Cecilia, & Pesole, Graziano. 2005. *Handbook of Comparative Genomics: Principles and Methodology*. Vol. 2. Wiley.

Sanger, F, Nicklen, S, & Coulson, A R. 1977a. DNA sequencing with chain-terminating inhibitors. *Proceedings of the National Academy of Sciences of the United States of America*, 74(12), 5463–7.

Sanger, F, Air, G M, Barrell, B G, Brown, N L, Coulson, A R, Fiddes, C A, Hutchison, C A, Slocombe, P M, & Smith, M. 1977b. Nucleotide sequence of bacteriophage phi X174 DNA. *Nature*, 265(5596), 687–95.

Shendure, Jay, & Ji, Hanlee. 2008. Next-generation DNA sequencing. *Nature biotechnology*, 26(10), 1135–45.

Smyth, Gordon K. 2004. Linear models and empirical bayes methods for assessing differential expression in microarray experiments. *Statistical applications in genetics and molecular biology*, 3(Jan.), Article3.

Soneson, Charlotte, & Delorenzi, Mauro. 2013. A comparison of methods for differential expression analysis of RNA-seq data. *BMC bioinformatics*, 14(Jan.), 91.

Storey, John D., Taylor, Jonathan E., & Siegmund, David. 2004. Strong control, conservative point estimation and simultaneous

conservative consistency of false discovery rates: a unified approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 66(1), 187–205.

Sultan, Marc, Schulz, Marcel H, Richard, Hugues, Magen, Alon, Klingenhoff, Andreas, Scherf, Matthias, Seifert, Martin, Borodina, Tatjana, Soldatov, Aleksey, Parkhomchuk, Dmitri, Schmidt, Dominic, O'Keeffe, Sean, Haas, Stefan, Vingron, Martin, Lehrach, Hans, & Yaspo, Marie-Laure. 2008. A global view of gene activity and alternative splicing by deep sequencing of the human transcriptome. *Science (New York, N.Y.)*, 321(5891), 956–60.

Tarazona, Sonia, García-Alcalde, Fernando, Dopazo, Joaquín, Ferrer, Alberto, & Conesa, Ana. 2011. Differential expression in RNA-seq: a matter of depth. *Genome research*, 21(12), 2213–23.

Trapnell, Cole, Pachter, Lior, & Salzberg, Steven L. 2009. TopHat: discovering splice junctions with RNA-Seq. *Bioinformatics (Oxford, England)*, 25(9), 1105–11.

Trapnell, Cole, Roberts, Adam, Goff, Loyal, Pertea, Geo, Kim, Daehwan, Kelley, David R, Pimentel, Harold, Salzberg, Steven L, Rinn, John L, & Pachter, Lior. 2012. Differential gene and transcript expression analysis of RNA-seq experiments with TopHat and Cufflinks. *Nature protocols*, 7(3), 562–78.

Trapnell, Cole, Hendrickson, David G, Sauvageau, Martin, Goff, Loyal, Rinn, John L, & Pachter, Lior. 2013. Differential analysis of gene regulation at transcript resolution with RNA-seq. *Nature biotechnology*, 31(1), 46–53.

Wagner, James R, Ge, Bing, Pokholok, Dmitry, Gunderson, Kevin L, Pastinen, Tomi, & Blanchette, Mathieu. 2010. Computational analysis of whole-genome differential allelic expression data in human. *PLoS computational biology*, 6(7).

Wang, Eric T, Sandberg, Rickard, Luo, Shujun, Khrebtukova, Irina, Zhang, Lu, Mayr, Christine, Kingsmore, Stephen F, Schroth, Gary P, & Burge, Christopher B. 2008a. Alternative isoform regulation in human tissue transcriptomes. *Nature*, 456(7221), 470–6.

Wang, Likun, Feng, Zhixing, Wang, Xi, Wang, Xiaowo, & Zhang, Xuegong. 2010. DEGseq: an R package for identifying differentially expressed genes from RNA-seq data. *Bioinformatics (Oxford, England)*, 26(1), 136–8.

Wang, Xu, Sun, Qi, McGrath, Sean D, Mardis, Elaine R, Soloway, Paul D, & Clark, Andrew G. 2008b. Transcriptome-wide identification of novel imprinted genes in neonatal mouse brain. *PloS one*, 3(12).

Wang, Zhong, Gerstein, Mark, & Snyder, Michael. 2009. RNA-Seq: a revolutionary tool for transcriptomics. *Nature Reviews Genetics*, 10(1), 57–63.

Yekutieli, Daniel, & Benjamini, Yoav. 2001. under dependency. *The Annals of Statistics*, 29(4), 1165–1188.

Yendrek, Craig R, Ainsworth, Elizabeth a, & Thimmapuram, Jyothi. 2012. The bench scientist's guide to statistical analysis of RNA-Seq data. *BMC research notes*, 5(1), 506.

Appendix A

R Scripts

The scripts corresponding to the data processing steps and the implementation of the two methods tested are shown below. The same procedure was run for each of the five remaining samples. Given that RNA-seq contains a lot of information and occupies a lot of memory space, the three scripts take several hours to run in a standard laptop/desktop computer. In particular, the data processing takes ≈ two hours to run while the two proportion comparison test and the Marascuilo procedure take ≈1.5 hours each per sample in an Apple MacBook Pro laptop.

A.1 Data processing

```
#-----------------DATA PROCESSING-------------------

#Reading the .bed data files
fileList<-
c("readsPerExon_rcc_caki1.bed","readsPerExon_rcc_caki2.bed","readsPerEx
on_rcc_ab.bed","readsPerExon_rcc_er.bed","readsPerExon_rcc_mf.bed","rea
dsPerExon_rcc_fg2.bed")

for(i in 1:length(fileList)){
assign(paste("data",i,sep=""),read.table(fileList[i], header=F))
}

#Gene list (with gene names)
```

```
geneList<-as.matrix(read.table("HumanGenes_hg19.txt", header=F))

geneDic<-cbind(geneList[,2],geneList[,13])

#Total number of mapped and unmapped reads per sample
totalReads<-read.table("totTagsNr.txt",header=T)

caki1.totalReads<-totalReads[6,3]
caki2.totalReads<-totalReads[7,3]
rccab.totalReads<-totalReads[3,3]
rccer.totalReads<-totalReads[1,3]
rccmf.totalReads<-totalReads[2,3]
rccfg2.totalReads<-totalReads[5,3]


#-------------------------------------------------------------
# Calculating exon length
exon.length<-data[,3]-data[,2] #Based on the data from Caki1


#Altering the matrix to separate genes and exons
library(stringr)
genEx<-str_split_fixed(data[,4], "_", 8)

genExMerge<-c(paste(genEx[,1],genEx[,2],sep="_"))

genEx2<-str_split_fixed(data2[,4], "_", 8)

genEx3<-str_split_fixed(data3[,4], "_", 8)

genEx4<-str_split_fixed(data4[,4], "_", 8)

genEx5<-str_split_fixed(data5[,4], "_", 8)

genEx6<-str_split_fixed(data6[,4], "_", 8)


#-------------------------------------------------------------
caki1<-data.frame(genExMerge,genEx[,4],genEx[,6],
data[,7])
colnames(caki1)<-c("gene","exon","chromossome",
```

```
"caki1")

caki2<-data.frame(genExMerge,genEx2[,4],genEx2[,6],
data2[,7])
colnames(caki2)<-c("gene","exon","chromossome",
"caki2")

rccab<-data.frame(genExMerge,genEx3[,4],genEx3[,6],
data3[,7])
colnames(rccab)<-c("gene","exon","chromossome",
"rccab")

rccer<-data.frame(genExMerge,genEx4[,4],genEx4[,6],
data4[,7])
colnames(rccer)<-c("gene","exon","chromossome",
"rccer")

rccmf<-data.frame(genExMerge,genEx5[,4],genEx5[,6],
data5[,7])
colnames(rccmf)<-c("gene","exon","chromossome",
"rccmf")

rccfg2<-data.frame(genExMerge,genEx6[,4],genEx6[,6],
data6[,7])
colnames(fg2)<-c("gene","exon","chromossome",
"rccfg2")

full<-data.frame(genExMerge,genEx[,4],genEx[,6],data[,7],
data2[,7],data3[,7],data4[,7],data5[,7],data6[,7])
colnames(full)<-c("gene","exon","chromossome",
"caki1","caki2","rccab","rccer","rccmf","rccfg2")


#-----------------------------------------------------
#Matching gene names with gene tags
geneTag<-paste(genEx[,1],genEx[,2],sep="_")

uni.geneDic<-geneDic[which(duplicated(geneDic[,1])==FALSE),]

matched<-rep(NA, times=nrow(myData)) #Genes with no gene tag will
remain as NA
```

```
for (i in 1:length(geneTag)){
if (geneTag[i] %in% uni.geneDic==TRUE){
matched[i]<-uni.geneDic[which(geneTag[i]==uni.geneDic[,1]),2]
print(i)
}
}

#-------------------------------------------------------------
#Wrote .CSV and re-read it to be able to work with the data
myData<-cbind(full,matched)
write.csv(myData, "table.csv")

#To re-read data
myData0<-read.csv("table.csv", header=T, sep=",")
myData0<-myData0[,2:11]
colnames(myData0)<-c("gene","exon","chromossome",
"caki1","caki2","rccab","rccer","rccmf","rccfg2","matched")

#-------------------------------------------------------------
#Index to distinguish genes (one number for each gene)
#index2
index2<-c(rep(1,1,nrow(myData0)))
for (i in 1:nrow(myData0)){
if (myData0$exon[i+1]>myData0$exon[i]) {index2[i+1]=index2[i]}
else
index2[i+1]=index2[i]+1
}



#Exons per gene
myData.exon<-myData0$exon+1
exon.number<-
unlist(tapply(myData.exon,index2,FUN=function(myData.exon){max(myData.e
xon)}))
head(exon.number)

exon.number2<-cbind(unique(index2),exon.number)

#-------------------------------------------------------------
##Removing the genes composed of 4 or less exons
```

```
#To identify genes composed of 1 exon
single.exon<-vector()
for (i in 1:(nrow(myData0)-1)){
if(myData0$exon[i+1]==myData0$exon[i]){
single.exon<-c(single.exon,index2[i])
}
}


removeSingleExon<-rep(1,times=nrow(myData))
for(i in 1:nrow(myData0)){
if(length(which((index2[i]==single.exon)==TRUE))==0){
removeSingleExon[i]=0
}
}

myData1<-myData0[-which(removeSingleExon==1),]
exon.length1<-exon.length[-which(removeSingleExon==1)]
index2.1<-index2[-which(removeSingleExon==1)]

#--------
#To identify genes composed of 4 or less exons
smallGene<-vector()
exon<-myData1$exon

s<-unlist(tapply(exon,index2.1,
FUN=function(exon){if(max(exon)<4) s<-index2.1[exon[max(exon)+1]]}))
head(s)
smallGene<-as.numeric(names(s))


#To verify that the genes to remove are correct (optional)
#v<-vector()
#for(i in 1:length(smallGene)){
#v<-c(v,length(which(index2.1==smallGene[i])))
#}


#To expand the vector with the small genes
removeSmallGenes<-rep(1,times=nrow(myData1))
for(i in 1:nrow(myData1)){
```

```
if(length(which((index2.1[i]==smallGene)==TRUE))==0){
removeSmallGenes[i]=0
}
}


myData2<-myData1[-(which(removeSmallGenes==1)),]
exon.length2<-exon.length1[-(which(removeSmallGenes==1))]
index3<-index2.1[-(which(removeSmallGenes==1))]


#---------------------------------------------------------------------
#To select transcriptionally active genes
#####caki1
caki1.sum<-myData2$caki1
caki1.sumReadsPerGene<-unlist(tapply(caki1.sum,index3,
FUN=function(caki1.sum){sum(caki1.sum)}))

quantile(caki1.sumReadsPerGene,1/3) #cutoff to consider a gene active

#Inactive genes
caki1.inactiveGenes<-
which(caki1.sumReadsPerGene<quantile(caki1.sumReadsPerGene,1/3))


#Expading the inactiveGenes vector according to the index
caki1.removeExons<-rep(1,times=length(caki1.sum))
for(i in 1:length(caki1.sum)){
if(length(which((index3[i]==caki1.inactiveGenes)==TRUE))==0){
caki1.removeExons[i]=0
}
}


#Removing the genes
caki1.final<-caki1.sum[-(which(caki1.removeExons==1))]
caki1.myData3<-myData2[-(which(caki1.removeExons==1)),]
caki1.exon.length3<-exon.length2[-(which(caki1.removeExons==1))]
caki1.index4<-index3[-(which(caki1.removeExons==1))]
```

```
#-----------------------------------------------------------
#To standardize the data
caki1.norm<-
(caki1.final)/((caki1.exon.length3/1000)*((caki1.totalReads)/1000000))

#-----------------------------------------------------------
#Estimating the reads based only on the more active genes
#caki1
quantile(caki1.sumReadsPerGene,2/3) #cutoff to consider a gene VERY
active

caki1.lessActiveGenes<-
which(caki1.sumReadsPerGene<quantile(caki1.sumReadsPerGene,2/3))


#Expanding the less active genes vector
caki1.lessActiveExons<-rep(1,times=length(caki1.norm))
for(i in 1:length(caki1.norm)){
if(length(which((caki1.index4[i]==caki1.lessActiveGenes)==TRUE))==0){
caki1.lessActiveExons[i]=0
}
}


caki1.totalReads.vector<-
rep(caki1.totalReads,times=length(exon.length))
caki1.totalReads.activeGenes<-caki1.totalReads.vector[-
which(removeSingleExon==1)]
caki1.totalReads.activeGenes1<-caki1.totalReads.activeGenes[-
(which(removeSmallGenes==1))]
caki1.totalReads.activeGenes2<-caki1.totalReads.activeGenes1[-
(which(caki1.removeExons==1))]
caki1.totalReads.activeGenes3<-caki1.totalReads.activeGenes2[-
(which(caki1.lessActiveExons==1))]


caki1.exonLengthForEstim<-caki1.exon.length3[-
(which(caki1.lessActiveExons==1))]


#Estimated number of reads after gene exclusion (both inactive gene and
```

```
less active genes (bottom 2/3 of the active genes))
caki1.estimReadsPerExon<-
(caki1.totalReads.activeGenes3/length(caki1.exonLengthForEstim))*caki1.
exonLengthForEstim


#Standardize estimReads - confirm
caki1.estimReadsPerExon.norm<-
(caki1.estimReadsPerExon)/((caki1.exonLengthForEstim/1000)*((caki1.tota
lReads)/1000000))


#Denominator for vector with proportions
caki1.estimReadsPerExon.norm.final<-
rep(caki1.estimReadsPerExon.norm[1],times=length(caki1.norm))

#Vector with proportions
caki1.propVector<-caki1.norm/caki1.estimReadsPerExon.norm.final

#Identifying genes with proportions>1
caki1.myData4<-cbind(caki1.myData3,caki1.index4)


#Index of genes with proportions>1
caki1.prop1.index<-
unique(caki1.myData4$caki1.index4[which(caki1.propVector>1)])

#Expanding the vector according of indexes of genes with proportions>1
to isolate
caki1.prop1.remove<-rep(1,times=nrow(caki1.myData4))
for(i in 1:nrow(caki1.myData4)){
if(length(which((caki1.myData4$caki1.index4[i]==caki1.prop1.index)==TRU
E))==0){
caki1.prop1.remove[i]=0
}
}

#Removing those genes of the data matrix and passing them to the
caki1.prop1.matrix
caki1.myData5<-caki1.myData4[-which(caki1.prop1.remove==1),]
```

```
#Matrix with only the genes with proportions bigger than 1
caki1.myData5.propBig1<-caki1.myData4[-which(caki1.prop1.remove!=1),]


#-------------------------------------------------------
#Adjusting Estim reads count and proportion vector without propBig
#eliminated the normalized read counts pertaining to genes with
proportions >1
caki1.norm2<-caki1.norm[-which(caki1.prop1.remove==1)]


#Expanding the less active genes vector for propBig
caki1.lessActiveExons.propBig<-rep(1,times=length(caki1.norm2))
for(i in 1:length(caki1.norm2)){
if(length(which((caki1.myData5$caki1.index4[i]==caki1.lessActiveGenes)=
=TRUE))==0){
caki1.lessActiveExons.propBig[i]=0
}
}


#Total reads for propBig
caki1.totalReads.activeGenes4<-
rep(caki1.totalReads.activeGenes3[1],times=length(caki1.exonLengthForEs
tim.propBig))

caki1.exon.length4<-caki1.exon.length3[-which(caki1.prop1.remove==1)]

caki1.exonLengthForEstim.propBig<-caki1.exon.length4[-
(which(caki1.lessActiveExons.propBig==1))]


#Estimated number of reads after gene exclusion (both inactive gene and
less active genes (bottom 2/3 of the active genes))
caki1.estimReadsPerExon.propBig<-
(caki1.totalReads.activeGenes4/length(caki1.exonLengthForEstim.propBig)
)*caki1.exonLengthForEstim.propBig


#Standardize estimReads
caki1.estimReadsPerExon.propBig.norm<-
```

```
(caki1.estimReadsPerExon.propBig)/((caki1.exonLengthForEstim.propBig/10
00)*((caki2.totalReads)/1000000))

#Denominator for vector with proportions
caki1.estimReadsPerExon.propBig.norm.final<-
rep(caki1.estimReadsPerExon.propBig.norm[1],times=length(caki1.norm2))


#Vector with proportions for analysis
caki1.propVector2<-
caki1.norm2/caki1.estimReadsPerExon.propBig.norm.final
```

A.2 Two proportions comparison test with the grouped BH correction for multiple testing

```
#--------------TWO PROPORTIONS COMPARISON TEST-------------

#1st vector - removing the last exon of each gene
#First, creating a vector with the number of exons of each
gene
caki1.combinProp<-
as.vector(tapply(caki1.myData5$exon,caki1.myData5$caki1.ind
ex4,max))+1


caki1.maxPosition<-rep(0,1,length(caki1.combinProp))
caki1.maxPosition[1]<-caki1.combinProp[1]
for(i in 1:(length(caki1.combinProp)-1)){
caki1.maxPosition[i+1]<-
caki1.maxPosition[i]+caki1.combinProp[i+1]
}

#1st vector of proportions (without the last exon of each
gene)
caki1.prop1<-caki1.propVector2[-caki1.maxPosition]
```

```
#2nd vector of proportions (without the first exon of each
gene)
caki1.prop2<-caki1.propVector2[-
which(caki1.myData5$exon==0)]


#-----------------------------------------------------------
#2 proportions comparison test
caki1.zStat<-rep(0,length(caki1.prop1))
caki1.pValue<-rep(0,length(caki1.prop1))

#Creating the vector with the adequate length os estimated
reads for Z-stat calculation
caki1.estimReadsPerExon.norm.zStat<-
rep(caki1.estimReadsPerExon.norm[1],times=length(caki1.prop
1))

#Calculating the Z-Stat
for(i in 1:length(caki1.prop1)){
caki1.zStat[i]<-(caki1.prop1[i]-
caki1.prop2[i])/(sqrt((caki1.prop1[i]*(1-
caki1.prop1[i])/caki1.estimReadsPerExon.norm.zStat[i])+(cak
i1.prop2[i]*(1-
caki1.prop2[i])/caki1.estimReadsPerExon.norm.zStat[i])))
if(caki1.prop1[i]==0 & caki1.prop2[i]==0){
caki1.zStat[i]=2.2e-16
}
}

#Calculating the p-value
caki1.pValue<-2*(1-pnorm(abs(caki1.zStat)))

#-----------------------------------------------------------
#Index for prop.test
caki1.group0<-caki1.myData5$caki1.index4[-
caki1.maxPosition]
caki1.group<-rep(0,times=length(caki1.group0))
for (i in 1:(length(caki1.group0)-1)){
if (caki1.group0[i]!=caki1.group0[i+1]){
caki1.group[i+1]=0
}
```

```
else {caki1.group[i+1]=caki1.group[i]+1}
}


#-----------------------------------------------------------
#Adjusting the p-value - Grouped Benjamini-Hochberg method
#library(structSSI)

caki1.pValueAdjGBH<-
Adaptive.GBH(caki1.pValue,caki1.group,alpha=0.05,method="ls
l")

#Creating output matrix
caki1.c2<-
c(as.numeric(caki1.pValueAdjGBH$rejected),as.numeric(caki1.
pValueAdjGBH$not.rejected))

caki1.c3<-caki1.pValueAdjGBH$adjp
caki1.c4<-cbind(caki1.c2,caki1.c3)

caki1.ordered<-caki1.c4[order(caki1.c4[,1]),]


caki1.gbh.matrix<-
cbind(caki1.group0,caki1.propTestStr,caki1.ordered[,1],caki
1.ordered[,1]+1,caki1.zStat,caki1.pValue)

#-----------------------------------------------------------
#Adjusting the p-value - Benjamini-Hochberg method
caki1.pValueAdj<-unlist(tapply(caki1.pValue,caki1.group0,
FUN=function(caki1.pValue){p.adjust(caki1.pValue,method="ho
chberg")}))


#Creating the output matrix
#Vector with binary significance (p-value=0.05)
caki1.prop.significance<-rep(0,length(caki1.pValueAdj))
for (i in 1:length(caki1.pValueAdj)){
if(caki1.pValueAdj[i]<0.05)
caki1.prop.significance[i]<-1
}
```

```
#Creates a string vector with the exon comparisons
#Vector for each exon
caki1.propTestStr<-paste("exon",caki1.group, " vs.
","exon",caki1.group+1,sep = " ")
head(caki1.propTestStr)

#Expression difference between pairs of exons
caki1.expressionDiff<-
as.vector(unlist(tapply(caki1.propVector2,caki1.myData5$cak
i1.index4,
FUN=function(caki1.propVector2){combn(caki1.propVector2,2)[
2,]-combn(caki1.propVector2,2)[1,]})))


caki1.bh.output.final<-
cbind(caki1.group0,caki1.propTestStr,caki1.group,caki1.grou
p+1,caki1.zStat,caki1.pValueAdj,caki1.expressionDiff,caki1.
prop.significance)


#Removing genes that do not have any significance=1
caki1.bh.sig<-as.numeric(caki1.bh.output.final[,8])
caki1.bh.sigSum<-
as.vector(unlist(tapply(caki1.bh.sig,as.numeric(caki1.bh.ou
tput.final[,1]),FUN=function(caki1.bh.sig){sum(caki1.bh.sig
)})))


caki1.bh.noItiGenes<-unique(caki1.bh.output.final[,1])[-
which(caki1.bh.sigSum>0)]

caki1.bh.iti.output<-caki1.bh.output.final[-
which(caki1.bh.output.final[,1] %in% caki1.bh.noItiGenes),]


#Number of exons in each of the selected genes
exonNum.bh.iti.index<-as.numeric(caki1.bh.iti.output[,4])
exonNum.bh.iti<-
as.vector(unlist(tapply(exonNum.bh.iti.index,as.numeric(cak
```

```
i1.bh.iti.output[,1]),FUN=function(exonNum.bh.iti.index){ma
x(exonNum.bh.iti.index)}))


#Expanding the number of exons vector to fit the output
matrix
exonNum.bh.iti.exp<-c()
for (i in 1:length(exonNum.bh.iti)){
exonNum.bh.iti.exp<-
c(exonNum.bh.iti.exp,rep(exonNum.bh.iti[i],times=exonNum.bh
.iti[i]))
}

caki1.bh.iti.output2<-
cbind(caki1.bh.iti.output,exonNum.bh.iti.exp)
head(caki1.bh.iti.output2)

#Selecting only the entries with significance=1
caki1.bh.iti.output3<-
caki1.bh.iti.output2[which(as.numeric(caki1.bh.iti.output2[
,8])==1),]


#Index for the first significance significance entry
caki1.bh.first.sig<-rep(1,times=nrow(caki1.bh.iti.output3))
for (i in 1:(length(caki1.bh.first.sig)-1)){
if(as.numeric(caki1.bh.iti.output3[i+1,1])==as.numeric(caki
1.bh.iti.output3[i,1])){
caki1.bh.first.sig[i+1]<-0
}
}


#Updating the output with the first.sig vector
caki1.bh.iti.output4<-
cbind(caki1.bh.iti.output3,caki1.bh.first.sig)


#Excluding the cases where the expression difference is
negative & first.sig=1
#--------Gene indexes to eliminate
```

```
caki1.bh.elimGene.index<-
as.numeric(caki1.bh.iti.output4[which(as.numeric(caki1.bh.i
ti.output4[,7])<0 & caki1.bh.first.sig==1),1])

caki1.bh.iti.output5<-caki1.bh.iti.output4[-
which(as.numeric(caki1.bh.iti.output4[,1]) %in%
caki1.bh.elimGene.index),]



#Eliminating first.sig
#caki1.elimGene.index2<-as.numeric()

caki1.bh.first.sig2<-
rep(1,times=nrow(caki1.bh.iti.output5))
for(i in 1:(nrow(caki1.bh.iti.output5)-1)){
if(as.numeric(caki1.bh.iti.output5[i+1,1])!=as.numeric(caki
1.bh.iti.output5[i,1]) &
as.numeric(caki1.bh.iti.output5[i+1,10])==0){
caki1.bh.first.sig2[i+1]<-0
}
}


#Addind total number of exons to the output matrix
caki1.bh.exon.number.iti<-c()
for (i in 1:nrow(caki1.bh.iti.output5)){
if (as.numeric(caki1.bh.iti.output5[i,1]) %in%
exon.number2[,1]){
caki1.bh.exon.number.iti[i]<-
exon.number2[which(exon.number2[,1]==which(exon.number2[,1]
%in% as.numeric(caki1.bh.iti.output5[i,1]))),2]
}
}


caki1.bh.iti.output6<-
cbind(caki1.bh.iti.output5,caki1.bh.exon.number.iti)


#cutoff
```

```
caki1.bh.cutoff<-(as.numeric(caki1.bh.iti.output6[,11])-
as.numeric(caki1.bh.iti.output6[,4]))/as.numeric(caki1.bh.i
ti.output6[,11])


caki1.bh.cutoff.index<-
caki1.bh.iti.output6[which(caki1.bh.iti.output6[,10]==1 &
caki1.bh.cutoff>0.6),1]

caki1.bh.iti.output7<-
caki1.bh.iti.output6[which(as.numeric(caki1.bh.iti.output6[
,1]) %in% as.numeric(caki1.bh.cutoff.index)),]

##############################################################
#######

#Separating simple ITI genes (caki1.iti.genes) from "more
complex" genes (caki1.spec.iti.genes)
caki1.bh.spec.iti.genes.index<-
as.numeric(caki1.bh.iti.output7[which(caki1.bh.iti.output7[
,10]==0),1])


#Only the ones with negative diff expression at at least on
pair of exons
caki1.bh.spec.iti.genes.index.neg<-
unique(caki1.bh.iti.output7[which(caki1.bh.iti.output7[,7]<
0),1])


caki1.bh.iti.genes<-caki1.bh.iti.output7[-
which(as.numeric(caki1.bh.iti.output7[,1]) %in%
unique(caki1.bh.spec.iti.genes.index.neg)),]

caki1.bh.spec.iti.genes<-
caki1.bh.iti.output7[which(as.numeric(caki1.bh.iti.output7[
,1]) %in% unique(caki1.bh.spec.iti.genes.index.neg)),]
head(caki1.bh.spec.iti.genes)
```

```
#Matching the gene names of the detected genes with ITI
#----------New Gene Dictionary
myGeneDic<-cbind(index2,matched)
uni.myGeneDic<-
myGeneDic[which(duplicated(myGeneDic[,1])==FALSE),]

#----------ITI genes (caki1.iti.genes)
caki1.bh.final.match<-rep(NA,
times=nrow(caki1.bh.iti.genes))
for (i in 1:nrow(caki1.bh.iti.genes)){
if (as.numeric(caki1.bh.iti.genes[i,1]) %in%
uni.myGeneDic[,1]==TRUE){
caki1.bh.final.match[i]<-
uni.myGeneDic[which(as.numeric(caki1.bh.iti.genes[i,1])==un
i.myGeneDic[,1]),2]
}
}

#----------"More complex" genes (caki1.spec.iti.genes)
caki1.bh.spec.final.match<-rep(NA,
times=nrow(caki1.bh.spec.iti.genes))
for (i in 1:nrow(caki1.bh.spec.iti.genes)){
if (as.numeric(caki1.bh.spec.iti.genes[i,1]) %in%
uni.myGeneDic[,1]==TRUE){
caki1.bh.spec.final.match[i]<-
uni.myGeneDic[which(as.numeric(caki1.spec.iti.genes[i,1])==
uni.myGeneDic[,1]),2]
}
}

caki1.bh.final.match<-rep(NA,
times=nrow(caki1.bh.iti.genes))
for (i in 1:nrow(caki1.bh.iti.genes)){
if (as.numeric(caki1.bh.iti.genes[i,1]) %in%
uni.myGeneDic[,1]==TRUE){
caki1.bh.final.match[i]<-
uni.myGeneDic[which(as.numeric(caki1.bh.iti.genes[i,1])==un
i.myGeneDic[,1]),2]
}
}
```

A.3 Marascuilo Procedure

```
#------------------MARASCUILO PROCEDURE-------------------

#Test statistic
caki1.testStat<-
unlist(tapply(caki1.propVector2,caki1.myData5$caki1.index4,
FUN=function(caki1.propVector2)
{abs(combn(caki1.propVector2,2)[1,]-combn(caki1.propVector2,2)[2,])}))


#Critical values calculation (rij)
#Combinations of propVector
caki1.combin<-
as.vector(tapply(caki1.myData5$exon,caki1.myData5$caki1.index4,max))+1
caki1.combinTotal<-cbind(caki1.combin,(caki1.combin*(caki1.combin-
1))/2)


#Vector corresponding to p1*(1-p1)/estRead
caki1.prop<-abs(caki1.propVector2*(1-
caki1.propVector2)/caki1.estimReadsPerExon.propBig.norm.final)


#Degrees of Freedom
caki1.degFreedom<-caki1.combinTotal[,1]-1
caki1.degFreedomExp<-rep(caki1.degFreedom,times=caki1.combinTotal[,2])


#Confidence level
confidence<-.95


#Computes the critical range value
caki1.range1<-sqrt(qchisq(confidence,caki1.degFreedomExp))
caki1.range2<-unlist(tapply(caki1.prop,caki1.myData5$caki1.index4,
FUN=function(caki1.prop){sqrt(combn(caki1.prop,2)[1,]+combn(caki1.prop,
2)[2,])}))

caki1.range<-caki1.range1*caki1.range2 #final
```

```
#Establishing significance (test statistic > critical value)
caki1.significance<-rep(0, length(caki1.testStat))
for (i in 1:length(caki1.testStat)){
if(caki1.testStat[i]>caki1.range[i]) caki1.significance[i]<-1
}


#Creates a string vector with the exon comparisons
   #Vector for each exon
caki1.propStr<-paste("exon",caki1.myData5$exon, sep = "")

#Vector with the sample length according to the index value
caki1.finalPropStr<-
unlist(tapply(caki1.propStr,caki1.myData5$caki1.index4,
FUN=function(caki1.propStr){paste(combn(caki1.propStr,2)[1,],combn(caki
1.propStr,2)[2,], sep=" vs. ")}))

#Expression difference between pairs of exons
caki1.expressionDiff<-
as.vector(unlist(tapply(caki1.propVector2,caki1.myData5$caki1.index4,
FUN=function(caki1.propVector2){combn(caki1.propVector2,2)[2,]-
combn(caki1.propVector2,2)[1,]})))

#Output of Marascuilo Procedure
caki1.output<-
cbind(caki1.finalPropStr,caki1.testStat,caki1.range,caki1.significance)


#Selecting the pairs of interest
caki1.myData5.exon<-caki1.myData5$exon
caki1.myData5.index<-caki1.myData5$caki1.index4
caki1.index<-
as.vector(unlist(tapply(caki1.myData5.index,caki1.myData5.index,
FUN=function(caki1.myData5.index){(combn(caki1.myData5.index,2)[1,])}))
)
head(caki1.index)
caki1.pairs1<-
as.vector(unlist(tapply(caki1.myData5.exon,caki1.myData5$caki1.index4,
```

```
FUN=function(caki1.myData5.exon){(combn(caki1.myData5.exon,2)[1,])})))
caki1.pairs2<-
as.vector(unlist(tapply(caki1.myData5.exon,caki1.myData5$caki1.index4,
FUN=function(caki1.myData5.exon){(combn(caki1.myData5.exon,2)[2,])})))
pairs3<-caki1.pairs2-caki1.pairs1
caki1.pairs4<-caki1.output[,4]

caki1.pairs<-
cbind(caki1.index,caki1.pairs1,caki1.pairs2,pairs3,caki1.pairs4)


#---------Establishing the consecutive pairs
caki1.consecutive.pairs<-rep(0,times=nrow(caki1.pairs))
for (i in 1:nrow(caki1.pairs)){
if(abs(caki1.pairs2[i]-caki1.pairs1[i])==1)
caki1.consecutive.pairs[i]<-1
}
caki1.pairsT<-caki1.pairs[-which(caki1.consecutive.pairs==0),]
caki1.output2<-caki1.output[-which(caki1.consecutive.pairs==0),]
caki1.finalPropStr.pairs<-caki1.finalPropStr[-
which(caki1.consecutive.pairs==0)]
caki1.expressionDiff2<-caki1.expressionDiff[-
which(caki1.consecutive.pairs==0)]


#Final output
caki1.output.final<-
cbind(as.numeric(caki1.pairsT[,1]),caki1.finalPropStr.pairs,caki1.pairs
T[,2:3],as.numeric(caki1.output2[,2]),as.numeric(caki1.output2[,3]),cak
i1.expressionDiff2,as.numeric(caki1.output2[,4]))
colnames(caki1.output.final)<-
c("index","comparisson","exon1","exon2","testStat","range","expressionD
iff","significance")


#Removing genes that do not have any significance=1
caki1.sig<-as.numeric(caki1.output.final[,8])
caki1.sigSum<-
as.vector(unlist(tapply(caki1.sig,as.numeric(caki1.output.final[,1]),FU
N=function(caki1.sig){sum(caki1.sig)})))
```

```
caki1.sigSum[which(unique(caki1.output.final[,1])==139)]
caki1.noItiGenes<-unique(caki1.output.final[,1])[-
which(caki1.sigSum>0)]


caki1.iti.output<-caki1.output.final[-which(caki1.output.final[,1] %in%
caki1.noItiGenes),]



#Number of exons in each of the selected genes
exonNum.iti.index<-as.numeric(caki1.iti.output[,4])
exonNum.iti<-
as.vector(unlist(tapply(exonNum.iti.index,as.numeric(caki1.iti.output[,
1]),FUN=function(exonNum.iti.index){max(exonNum.iti.index)})))


#Expanding the number of exons vector to fit the output matrix
exonNum.iti.exp<-c()
for (i in 1:length(exonNum.iti)){
exonNum.iti.exp<-
c(exonNum.iti.exp,rep(exonNum.iti[i],times=exonNum.iti[i]))
}

caki1.iti.output2<-cbind(caki1.iti.output,exonNum.iti.exp)


#Selecting only the entries with significance=1
caki1.iti.output3<-
caki1.iti.output2[which(as.numeric(caki1.iti.output2[,8])==1),]


#Index for the first significance entry
caki1.first.sig<-rep(1,times=nrow(caki1.iti.output3))
for (i in 1:(length(first.sig)-1)){
if(as.numeric(caki1.iti.output3[i+1,1])==as.numeric(caki1.iti.output3[i
,1])){
caki1.first.sig[i+1]<-0
}
}
```

```
#Updating the output with the first.sig vector
caki1.iti.output4<-cbind(caki1.iti.output3,first.sig)


#Excluding the cases where the expression differecence is negative &
first.sig=1



#--------Gene indexes to eliminate
caki1.elimGene.index<-
as.numeric(caki1.iti.output4[which(as.numeric(caki1.iti.output4[,7])<0
& first.sig==1),1])

caki1.iti.output5<-caki1.iti.output4[-
which(as.numeric(caki1.iti.output4[,1]) %in% caki1.elimGene.index),]


#####################################################################

#Addind total number of exons to the output matrix
caki1.exon.number.iti<-c()
for (i in 1:nrow(caki1.iti.output5)){
if (as.numeric(caki1.iti.output5[i,1]) %in% exon.number2[,1]){
caki1.exon.number.iti[i]<-
exon.number2[which(exon.number2[,1]==which(exon.number2[,1] %in%
as.numeric(caki1.iti.output5[i,1]))),2]
}
}


caki1.iti.output6<-cbind(caki1.iti.output5,caki1.exon.number.iti)


caki1.cutoff<-(as.numeric(caki1.iti.output6[,11])-
as.numeric(caki1.iti.output6[,4]))/as.numeric(caki1.iti.output6[,11])


caki1.cutoff2<-rep(0,times=length(caki1.cutoff))
caki1.cutoff2[1]<-caki1.cutoff[1]
```

```
for (i in 1:(caki1.cutoff-1)){
if
(as.numeric(caki1.iti.output6[i+1,1])==as.numeric(caki1.iti.output6[i,1
]))
caki1.cutoff2[i+1]<-caki1.cutoff[i]
}


caki1.cutoff.index<-caki1.iti.output6[which(caki1.iti.output6[,10]==1 &
caki1.cutoff>0.6),1]

caki1.iti.output7<-
caki1.iti.output6[which(as.numeric(caki1.iti.output6[,1]) %in%
as.numeric(caki1.cutoff.index)),]

#####################################################################

#Separating simple ITI genes (caki1.iti.genes)
caki1.spec.iti.genes.index<-
as.numeric(caki1.iti.output7[which(caki1.iti.output7[,10]==0),1])


#Only the ones with negative diff expression at at least on pair of
exons
caki1.spec.iti.genes.index.neg<-
unique(caki1.iti.output7[which(caki1.iti.output7[,7]<0),1])


caki1.iti.genes<-caki1.iti.output7[-
which(as.numeric(caki1.iti.output7[,1]) %in%
unique(caki1.spec.iti.genes.index.neg)),]


caki1.spec.iti.genes<-
caki1.iti.output7[which(as.numeric(caki1.iti.output7[,1]) %in%
unique(caki1.spec.iti.genes.index.neg)),]


#Matching the gene names of the detected genes with ITI
#----------New Gene Dictionary
myGeneDic<-cbind(index2,matched)
```

```
uni.myGeneDic<-myGeneDic[which(duplicated(myGeneDic[,1])==FALSE),]

#----------ITI genes (caki1.iti.genes)
caki1.final.match<-rep(NA, times=nrow(caki1.iti.genes))
for (i in 1:nrow(caki1.iti.genes)){
if (as.numeric(caki1.iti.genes[i,1]) %in% uni.myGeneDic[,1]==TRUE){
caki1.final.match[i]<-
uni.myGeneDic[which(as.numeric(caki1.iti.genes[i,1])==uni.myGeneDic[,1]
),2]
}
}


#----------"More complex" genes (caki1.spec.iti.genes
caki1.spec.final.match<-rep(NA, times=nrow(caki1.spec.iti.genes))
for (i in 1:nrow(caki1.spec.iti.genes)){
if (as.numeric(caki1.spec.iti.genes[i,1]) %in%
uni.myGeneDic[,1]==TRUE){
caki1.spec.final.match[i]<-
uni.myGeneDic[which(as.numeric(caki1.spec.iti.genes[i,1])==uni.myGeneDi
c[,1]),2]
}
}


#Final gene list
caki1.final.match<-rep(NA, times=nrow(caki1.iti.genes))
for (i in 1:nrow(caki1.iti.genes)){
if (as.numeric(caki1.iti.genes[i,1]) %in% uni.myGeneDic[,1]==TRUE){
caki1.final.match[i]<-
uni.myGeneDic[which(as.numeric(caki1.iti.genes[i,1])==uni.myGeneDic[,1]
),2]
}
}
```